

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2002-24039

(P2002-24039A)

(43)公開日 平成14年1月25日 (2002.1.25)

(51)Int.Cl.<sup>7</sup>

G 06 F 9/54  
9/445  
11/34

識別記号

F I

G 06 F 11/34  
9/06

テーマコード(参考)

L 5 B 0 4 2  
6 4 0 C 5 B 0 7 6  
6 1 0 Q

審査請求 未請求 請求項の数14 OL (全27頁)

(21)出願番号 特願2000-211446(P2000-211446)

(22)出願日 平成12年7月12日 (2000.7.12)

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 前川 隆昭

東京都千代田区丸の内二丁目2番3号 三菱電機株式会社内

(72)発明者 西岡 篤史

東京都千代田区丸の内二丁目2番3号 三菱電機株式会社内

(74)代理人 100057874

弁理士 曾我 道照 (外6名)

Fターム(参考) 5B042 GA12 KK14 MC40

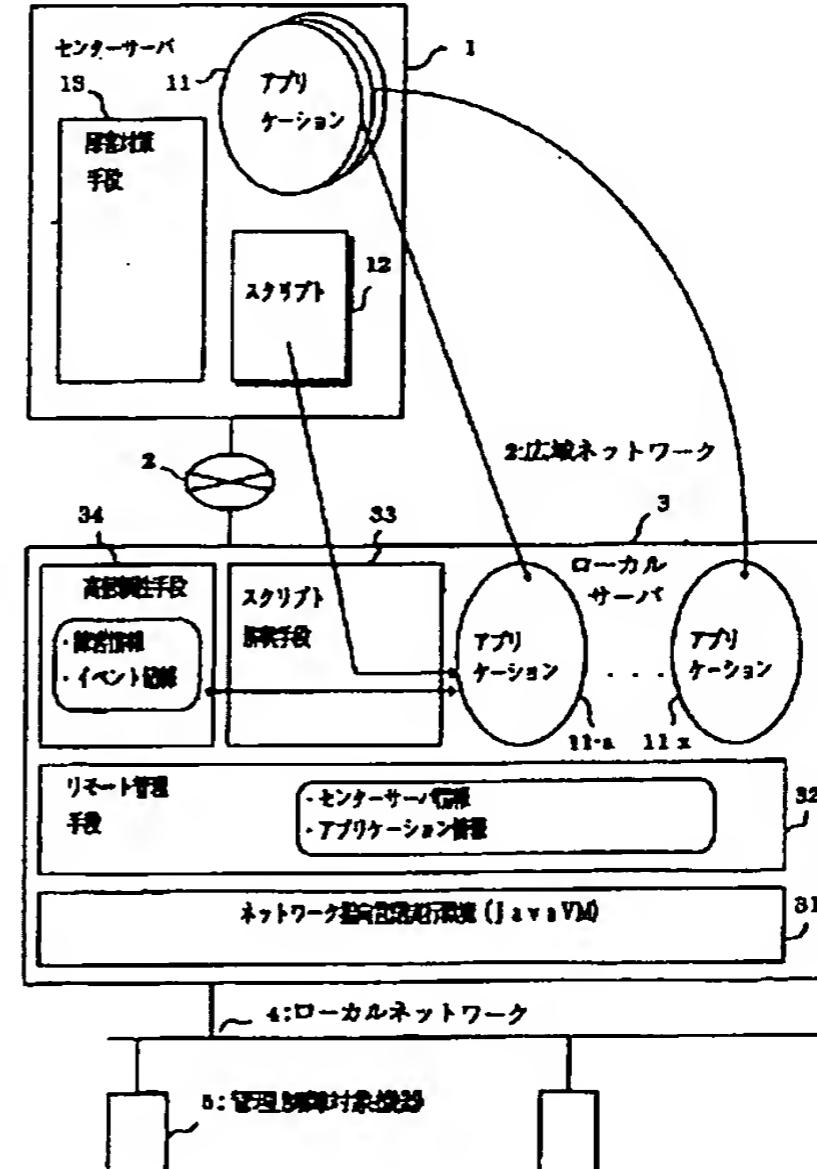
5B076 BB06 CA01 CA04

(54)【発明の名称】 ソフトウェア管理システム

(57)【要約】

【課題】 広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムを得る。

【解決手段】 センターサーバ1は、アプリケーション11の動作を記述したスクリプト12と、障害発生時の対応を行う障害対策手段13とを備え、ローカルサーバ3は、ネットワーク指向言語実行環境31と、アプリケーションのダウンロードや削除を行うリモート管理手段32と、スクリプト解釈手段33と、アプリケーションの実行中に発生するイベント情報の記録や、障害発生時の情報管理および復旧処理を行う高信頼性手段34とを備え、広域ネットワーク2を介してダウンロードされたソフトウェアの安全確実な動作を保証するとともに、万一異常が発生しても障害情報収集や復旧を支援する機能を提供した。



## 【特許請求の範囲】

【請求項1】 センターサーバと、広域ネットワークを介して前記センターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、  
前記センターサーバは、  
前記ローカルサーバにダウンロードさせて動作するアプリケーションと、  
前記アプリケーションの動作を記述したスクリプトと、  
障害発生時の対応を行う障害対策手段とを備え、  
前記ローカルサーバは、  
ネットワーク指向言語実行環境と、  
前記アプリケーションの前記センターサーバからのダウンロードや処理終了後の削除などを行うリモート管理手段と、  
前記スクリプトを解釈して前記アプリケーションに処理を要求するスクリプト解釈手段と、  
前記アプリケーションの実行中に発生するイベント情報の記録や、前記障害発生時の情報管理および復旧処理を行う高信頼性手段とを備えたことを特徴とするソフトウェア管理システム。

【請求項2】 前記リモート管理手段は、  
センターサーバ情報およびアプリケーション情報と、  
前記アプリケーションの実行要求に対応する要求処理手段と、  
前記センターサーバ情報および前記アプリケーション情報に基づいて前記アプリケーションを前記センターサーバからダウンロードするアプリケーションダウンロード手段と、  
前記アプリケーション情報に基づいて前記アプリケーションの起動や削除などの処理を行うアプリケーション管理手段とを含むことを特徴とする請求項1に記載のソフトウェア管理システム。

【請求項3】 前記スクリプト解釈手段は、  
スクリプト定義およびイベント一覧と、  
前記スクリプト定義にしたがって前記スクリプトを解釈して前記スクリプトの定義内容に相当するイベントを出力する解釈手段と、  
前記イベントを取り込むとともに前記イベント一覧にしたがって前記イベントにより駆動される処理を抽出するイベント駆動手段とを含むことを特徴とする請求項1に記載のソフトウェア管理システム。

【請求項4】 前記スクリプトは、XML (eXtensible Markup Language) により記述され、  
前記スクリプト定義は、DTD (Document Type Definition) により記述されたことを特徴とする請求項1に記載のソフトウェア管理システム。

【請求項5】 前記障害対策手段は、

前記障害発生時の障害情報を前記ローカルサーバから取得する障害情報取得手段と、  
前記障害発生時の障害対策を決定して前記ローカルサーバに通知する障害対策通知手段とを含み、  
前記高信頼性手段は、  
前記障害発生時に障害の発生を検知する障害検知手段と、  
前記障害発生時の障害情報を収集する障害情報収集手段と、

10 前記障害情報を前記センターサーバに通知する障害通知手段と、  
前記センターサーバからの障害対策に応じて前記障害を復旧させる復旧手段と、  
前記イベント情報を収集記録するイベント収集手段とを含むことを特徴とする請求項1に記載のソフトウェア管理システム。

【請求項6】 前記障害対策手段は、前記障害情報の種別毎に障害対策を格納する障害対策一覧表を含み、  
前記障害対策通知手段は、

20 前記障害情報の種別に対応した障害対策を前記障害対策一覧表に基づいて検出する障害対策検出手段と、  
前記障害対策を前記ローカルサーバに通知する通知手段とを含むことを特徴とする請求項5に記載のソフトウェア管理システム。

【請求項7】 前記障害対策手段は、前記障害発時の障害情報を前記ローカルサーバから取得する障害情報取得手段を含み、  
前記高信頼性手段は、  
前記障害発時に障害の発生を検知する障害検知手段と、

30 前記障害発時の障害情報を収集する障害情報収集手段と、  
前記障害発時の障害への対応を自律的に行い前記障害を自動復旧させる復旧手段と、  
前記障害情報および自動復旧の情報を前記センターサーバに通知する通知手段と、  
前記イベント情報を収集記録するイベント収集手段とを含むことを特徴とする請求項1に記載のソフトウェア管理システム。

40 【請求項8】 前記ローカルサーバは複数存在し、それぞれ、前記ネットワーク指向言語実行環境、前記リモート管理手段、前記スクリプト解釈手段および前記高信頼性手段を備えたことを特徴とする請求項1に記載のソフトウェア管理システム。

【請求項9】 前記センターサーバは複数存在し、それぞれ、前記アプリケーション、前記スクリプトおよび前記障害対策手段を備えたことを特徴とする請求項8に記載のソフトウェア管理システム。

【請求項10】 前記ローカルサーバは複数存在し、  
50 少なくとも1つのローカルサーバは、前記ネットワーク

指向言語実行環境、前記リモート管理手段、前記スクリプト解釈手段および前記高信頼性手段を備え、

他のローカルサーバは、前記ネットワーク指向言語実行環境、前記リモート管理手段および前記スクリプト解釈手段を備えたことを特徴とする請求項1に記載のソフトウェア管理システム。

【請求項11】センターサーバと、広域ネットワークを介して前記センターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、

前記センターサーバは、

前記ローカルサーバにダウンロードさせて動作するアプリケーションと、

前記アプリケーションの動作を記述したスクリプトとを備え、

前記ローカルサーバは、

ネットワーク指向言語実行環境と、

前記アプリケーションの前記センターサーバからのダウンロードや処理終了後の削除などを行うリモート管理手段と、

前記スクリプトを解釈して前記アプリケーションに処理を要求するスクリプト解釈手段とを備えたことを特徴とするソフトウェア管理システム。

【請求項12】前記ローカルサーバは複数存在し、それぞれ、前記ネットワーク指向言語実行環境、前記リモート管理手段および前記スクリプト解釈手段を備えたことを特徴とする請求項11に記載のソフトウェア管理システム。

【請求項13】センターサーバと、広域ネットワークを介して前記センターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、

前記センターサーバは、

前記ローカルサーバにダウンロードさせて動作するアプリケーションと、

障害発生時の対応を行う障害対策手段とを備え、

前記ローカルサーバは、

ネットワーク指向言語実行環境と、

前記アプリケーションの前記センターサーバからのダウンロードや処理終了後の削除などを行うリモート管理手段と、

前記アプリケーションの実行中に発生するイベント情報の記録や、前記障害発生時の情報管理および復旧処理を行う高信頼性手段とを備えたことを特徴とするソフトウェア管理システム。

【請求項14】前記ローカルサーバは複数存在し、それぞれ、前記ネットワーク指向言語実行環境、前記リモート管理手段および前記高信頼性手段を備えたことを特徴とする請求項13に記載のソフトウェア管理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、センターサーバと、広域ネットワークを介してセンターサーバに接続されたローカルサーバとを備え、センターサーバが広域ネットワーク全体を管理するように構成されたネットワークシステムからなるソフトウェア管理システムに関し、特にセンターサーバからローカルサーバへのアプリケーションソフトウェアのダウンロード機能および信頼性を向上させたソフトウェア管理システムに関するものである。

【0002】

【従来の技術】近年、インターネットなどの広域ネットワークの利用がますます進みつつあり、また、ネットワーク指向のプログラム言語である「Java（登録商標）」の普及とともに、広域ネットワークを介してセンターサーバからローカルサーバ（コンピュータなど）にソフトウェアをダウンロードする処理が増えてきている。

【0003】しかも、最近は、組込み型のローカルサーバのように、メモリ容量の比較的小ないコンピュータにおいても、アプリケーションソフトウェアをダウンロードする仕組みが必要になってきている。

【0004】しかしながら、メモリ容量の少ない組込み型のローカルサーバにおいては、全てのアプリケーションを常時メモリ上に置くことができない場合もあり、また、通常、ディスクなどの補助記憶装置も装備されていないので、必要に応じてソフトウェアをダウンロードし、不要になれば削除する機能が必要となる。

【0005】従来より、上記のようなりモードでのソフトウェア管理機能は、たとえばOGSi（Open Gateway Service Initiative）が定めたOSG（Open Gateway Service）などの規格がある（詳細は、OGSiのホームページ「<http://www.osgi.org/>」を参照）。

【0006】上記OSG規格は、ソフトウェアのダウンロードや削除の方式を規定しているが、さらに高い信頼性が必要なシステムに適用する場合には、独自に、高信頼性機能や安全確実な動作を保証するための仕組みを実装する必要がある。

【0007】たとえば特開平11-65968号公報に記載された「ネットワーク管理方式およびそのシステム」においては、ITU-Tの勧告「M.3010」で規定されたマネージャ・エージェントモデルに基づく方式に、Javaで記述されたソフトウェアをダウンロードする仕組みを加えたシステムが提案されている。

【0008】上記公報に記載のネットワーク管理方式は、ITU-Tの勧告「M.3010」で規定されているマネージャ・エージェントモデルに基づく方式を前提としているが、Javaで記述されたソフトウェア自体

の動作が異常となった場合の処置や、信頼性を高めるための工夫については何ら考慮していない。

【0009】すなわち、従来のネットワークを介したソフトウェアのリモート管理システムにおいては、いずれも、ダウンロードや削除などの基本的な機能については規定しているものの、ダウンロードされたソフトウェアが異常な動作をしたときの処理や、安全確実な動作を保証する仕組みなど、システムとしての信頼性を高める機能については何ら考慮されていない。

【0010】今後、ネットワークを介したソフトウェアの流通がますます進むことを考慮すると、ダウンロードしたソフトウェアの安全性や、万一の異常動作時に対処することのニーズが強いことは言うまでもない。しかしながら、従来システムでは、十分にその要求に応えることはできていない。

【0011】

【発明が解決しようとする課題】従来のソフトウェア管理システムは以上のように、ダウンロードしたソフトウェアの安全性を確保することや、万一の異常動作時の対処することなどの要求に応えることができないという問題点があった。

【0012】この発明は上記のような問題点を解決するためになされたもので、ネットワークを介してダウンロードされたソフトウェアの安全確実な動作を保証するとともに、万一異常が発生しても障害情報収集や復旧を支援する機能を提供し、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムを得ることを目的とする。

【0013】

【課題を解決するための手段】この発明の請求項1に係るソフトウェア管理システムは、センターサーバと、広域ネットワークを介してセンターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、センターサーバは、ローカルサーバにダウンロードさせて動作するアプリケーションと、アプリケーションの動作を記述したスクリプトと、障害発生時の対応を行う障害対策手段とを備え、ローカルサーバは、ネットワーク指向言語実行環境と、アプリケーションのセンターサーバからのダウンロードや処理終了後の削除などを行うリモート管理手段と、スクリプトを解釈してアプリケーションに処理を要求するスクリプト解釈手段と、アプリケーションの実行中に発生するイベント情報の記録や、障害発生時の情報管理および復旧処理を行う高信頼性手段とを備えたものである。

【0014】また、この発明の請求項2に係るソフトウェア管理システムは、請求項1において、リモート管理手段は、センターサーバ情報およびアプリケーション情報と、アプリケーションの実行要求に対応する要求処理手段と、センターサーバ情報およびアプリケーション情

報に基づいてアプリケーションをセンターサーバからダウンロードするアプリケーションダウンロード手段と、アプリケーション情報に基づいてアプリケーションの起動や削除などの処理を行うアプリケーション管理手段とを含むものである。

【0015】また、この発明の請求項3に係るソフトウェア管理システムは、請求項1において、スクリプト解釈手段は、スクリプト定義およびイベント一覧と、スクリプト定義にしたがってスクリプトを解釈してスクリプトの定義内容に相当するイベントを出力する解釈手段と、イベントを取り込むとともにイベント一覧にしたがってイベントにより駆動される処理を抽出するイベント駆動手段とを含むものである。

【0016】また、この発明の請求項4に係るソフトウェア管理システムは、請求項1において、スクリプトは、XML (eXtensible Markup Language) により記述され、スクリプト定義は、DTD (Document Type Definition) により記述されたものである。

【0017】また、この発明の請求項5に係るソフトウェア管理システムは、請求項1において、障害対策手段は、障害発生時の障害情報をローカルサーバから取得する障害情報取得手段と、障害発生時の障害対策を決定してローカルサーバに通知する障害対策通知手段とを含み、高信頼性手段は、障害発生時に障害の発生を検知する障害検知手段と、障害発生時の障害情報を収集する障害情報収集手段と、障害情報をセンターサーバに通知する障害通知手段と、センターサーバからの障害対策に応じて障害を復旧させる復旧手段と、イベント情報を収集

記録するイベント収集手段とを含むものである。

【0018】また、この発明の請求項6に係るソフトウェア管理システムは、請求項5において、障害対策手段は、障害情報の種別毎に障害対策を格納する障害対策一覧表を含み、障害対策通知手段は、障害情報の種別に対応した障害対策を障害対策一覧表に基づいて検出する障害対策検出手段と、障害対策をローカルサーバに通知する通知手段とを含むものである。

【0019】また、この発明の請求項7に係るソフトウェア管理システムは、請求項1において、障害対策手段は、障害発生時の障害情報をローカルサーバから取得する障害情報取得手段を含み、高信頼性手段は、障害発生時に障害の発生を検知する障害検知手段と、障害発生時の障害情報を収集する障害情報収集手段と、障害発生時の障害への対応を自律的に行い障害を自動復旧させる復旧手段と、障害情報および自動復旧の情報をセンターサーバに通知する通知手段と、イベント情報を収集記録するイベント収集手段とを含むものである。

【0020】また、この発明の請求項8に係るソフトウェア管理システムは、請求項1において、ローカルサーバは複数存在し、それぞれ、ネットワーク指向言語実行

環境、リモート管理手段、スクリプト解釈手段および高信頼性手段を備えたものである。

【0021】また、この発明の請求項9に係るソフトウェア管理システムは、請求項8において、センターサーバは複数存在し、それぞれ、アプリケーション、スクリプトおよび障害対策手段を備えたものである。

【0022】また、この発明の請求項10に係るソフトウェア管理システムは、請求項1において、ローカルサーバは複数存在し、少なくとも1つのローカルサーバは、ネットワーク指向言語実行環境、リモート管理手段、スクリプト解釈手段および高信頼性手段を備え、他のローカルサーバは、ネットワーク指向言語実行環境、リモート管理手段およびスクリプト解釈手段を備えたものである。

【0023】また、この発明の請求項11に係るソフトウェア管理システムは、センターサーバと、広域ネットワークを介してセンターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、センターサーバは、ローカルサーバにダウンロードさせて動作するアプリケーションと、アプリケーションの動作を記述したスクリプトとを備え、ローカルサーバは、ネットワーク指向言語実行環境と、アプリケーションのセンターサーバからのダウンロードや処理終了後の削除などをを行うリモート管理手段と、スクリプトを解釈してアプリケーションに処理を要求するスクリプト解釈手段とを備えたものである。

【0024】また、この発明の請求項12に係るソフトウェア管理システムは、請求項11において、ローカルサーバは複数存在し、それぞれ、ネットワーク指向言語実行環境、リモート管理手段およびスクリプト解釈手段を備えたものである。

【0025】また、この発明の請求項13に係るソフトウェア管理システムは、センターサーバと、広域ネットワークを介してセンターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、センターサーバは、ローカルサーバにダウンロードさせて動作するアプリケーションと、障害発生時の対応を行う障害対策手段とを備え、ローカルサーバは、ネットワーク指向言語実行環境と、アプリケーションのセンターサーバからのダウンロードや処理終了後の削除などをを行うリモート管理手段と、アプリケーションの実行中に発生するイベント情報の記録や、障害発生時の情報管理および復旧処理を行う高信頼性手段とを備えたものである。

【0026】また、この発明の請求項14に係るソフトウェア管理システムは、請求項13において、ローカルサーバは複数存在し、それぞれ、ネットワーク指向言語実行環境、リモート管理手段および高信頼性手段を備えたものである。

【0027】

【発明の実施の形態】実施の形態1. 以下、図面を参照しながら、この発明の実施の形態1について詳細に説明する。図1はこの発明の実施の形態1を概念的に示すブロック構成図である。

【0028】図1において、センターサーバ1は、アプリケーション11と、アプリケーションの動作を記述したスクリプト12と、障害発生時の対応を行う障害対策手段13とを有する。

【0029】アプリケーション11は、後述するように、複数のアプリケーション11a～11xを含み、スクリプト12は、各アプリケーション11a～11xに応じた複数のスクリプト12a～12xを含む。

【0030】広域ネットワーク2は、センターサーバ1とローカルサーバ3とを接続している。センターサーバ1内のアプリケーション11は、広域ネットワーク2を介してローカルサーバ3にダウンロードされて動作する。

【0031】ローカルサーバ3は、必要に応じてセンターサーバ1からダウンロードされるアプリケーション11a～11xと、Java VMからなるネットワーク指向言語実行環境31と、リモート管理手段32と、スクリプト解釈手段33と、高信頼性手段34とを有する。

【0032】リモート管理手段32は、アプリケーション11のセンターサーバ1からのダウンロードや処理終了後の削除などの処理を実行する。高信頼性手段34は、アプリケーション11の実行中に発生するイベント情報の記録や、障害発生時の情報管理および復旧処理を行う。

【0033】ローカルネットワーク4は、ローカルサーバ3と管理制御対象機器5とを接続している。

【0034】次に、図1に示したこの発明の実施の形態1によるシステム全体の動作について説明する。まず、センターサーバ1内にあらかじめ登録されているアプリケーション11のうち、ローカルサーバ3に必要なアプリケーション11a～11xは、広域ネットワーク2を経由してローカルサーバ3にダウンロードされる。

【0035】ダウンロードされたアプリケーション11a～11xは、ローカルネットワーク4を介して接続された管理制御対象機器5に対し、機器の制御や機器内の情報の取得などの処理を行う。

【0036】次に、図2のフローチャートを参照しながら、図1に示したこの発明の実施の形態1による具体的な動作について説明する。すなわち、ダウンロードされたソフトウェアの安全で確実な動作の仕組みと、万一の異常発生時の対応の仕組みとについて説明する。

【0037】なお、ここで、センターサーバ1からダウンロードして動作するネットワーク指向のソフトウェア実行環境31には、現在もっとも普及している「Java」の実行環境「Java VM (Java Virtual Machine)」を想定している。

【0038】ただし、同種の機能を持つ環境であれば、「Java VM」以外のものでも、同等の効果を奏することは言うまでもない。

【0039】図2において、まず、ローカルサーバ3内のリモート管理手段32により、センターサーバ1からアプリケーション11をローカルサーバ3にダウンロードする(ステップS1)。

【0040】ダウンロードされたアプリケーション11は、スクリプト解釈手段33から処理要求が発生するまでは、待機状態になる(ステップS2)。

【0041】ここで、アプリケーション11に対する処理要求が発生し、ステップS2において処理要求有り(すなわち、YES)と判定され、この処理要求がリモート管理手段32に入力されると、この時点からアプリケーション11に起動がかかる(ステップS3)。

【0042】このとき、処理要求に対応したスクリプト12は、センターサーバ1からローカルサーバ3にダウンロードされ、ローカルサーバ3内のスクリプト解釈手段33によって、その処理内容が解釈される。

【0043】また、スクリプト解釈手段33は、解釈された処理内容にしたがう処理要求をアプリケーション11に送り、処理を実行させる(ステップS4)。このように、アプリケーション11を実行させるためにスクリプト12を仲介させるのは、安全で確実な動作を実行するためである。

【0044】すなわち、アプリケーション11は、センターサーバ1への登録時に動作検証が実行されるが、このとき、アプリケーション11の動作をスクリプト12により規定しておけば、動作範囲が決定し、スクリプト12に記載されている範囲外の動作は実行されない。

【0045】また、あらかじめ、全ての動作範囲をスクリプト12で規定しておけば、動作検証はスクリプト12に記載の範囲のみで実行すればよく、効率的で確実な検証を実行することができる。

【0046】したがって、あらかじめ、全ての動作の検証を実行することができ、アプリケーション11の実行時の安全性を保証することができる。また、スクリプト12に記載された動作のみが実行されるので、仕様が明確になり、確実に必要な動作を確認することができる。

【0047】処理要求が発生したアプリケーション11の動作中においては、アプリケーション11自体から、または、OSなどのシステムから、種々のイベントが発行される。

【0048】図2に戻り、高信頼性手段34は、発行されたイベントを検知し、イベント情報として内部に記録する(ステップS5)。

【0049】続いて、高信頼性手段34は、発行されたイベントに関して障害が発生したか否かを判定し(ステップS6)、障害発生のイベントである(すなわち、YES)と判定された場合には、イベント情報の蓄積処理

(ステップS5)のみでなく、所定の手続きにしたがって障害への対応処理を実行し(ステップS7)、ステップS8に進む。

【0050】また、ステップS6において、イベントに障害がない(すなわち、NO)と判定されれば、直ちにステップS8に進む。ステップS8においては、全ての処理が終了したか否かを判定し、全ての処理が終了した(すなわち、YES)と判定されれば、図2の処理ルーチンを終了して、次のアプリケーション11への処理要求を待つ。

【0051】一方、ステップS8において、全ての処理が終了していない(すなわち、NO)と判定されれば、イベント情報の記録ステップS5に戻り、上記処理を繰り返す。

【0052】このように、高信頼性手段34は、動作中のアプリケーション11が発行するイベント情報を記録すること(ステップS5)により、アプリケーション11の異常を検知して必要な対応処置(ステップS7)を実行する。この機能により、動作中に万一異常が起こっても適切な対応処置が行われ、システムとしての機能が損なわれることはない。

【0053】実施の形態2. なお、上記実施の形態1では、ローカルサーバ3内のリモート管理手段32について詳述しなかったが、リモート管理手段32を図3のように構成してもよい。

【0054】以下、図3を参照しながら、リモート管理手段32を具体化したこの発明の実施の形態2について説明する。

【0055】図3はローカルサーバ3内のリモート管理手段32の機能を具体的に示すブロック構成図であり、前述(図1参照)と同様のものについては、同一符号を付して詳述を省略する。

【0056】図3において、ローカルネットワーク4および管理制御対象機器5などの、リモート管理手段32と直接関連しない構成は省略されている。また、センターサーバ1においては、スクリプト12aおよびアプリケーション11aのみが代表的に示されている。

【0057】リモート管理手段32は、センターサーバ情報321と、アプリケーション情報322と、要求処理手段323と、アプリケーションダウンロード手段324と、アプリケーション管理手段325とを備えている。

【0058】要求処理手段323は、スクリプト解釈手段33から発生されたアプリケーション11の実行要求に対応する。

【0059】アプリケーションダウンロード手段324は、センターサーバ情報321およびアプリケーション情報322に基づいて、アプリケーション11をセンターサーバ1からダウンロードする。

【0060】アプリケーション管理手段325は、アブ

リケーション情報322に基づいて、アプリケーション11の起動や削除などの処理を実行する。

【0061】次に、図4のフローチャートを参照しながら、図3に示したこの発明の実施の形態2による具体的な動作について説明する。なお、システムとしての動作は前述と同様なので、ここでは、リモート管理手段32内の処理動作のみに注目して説明する。

【0062】まず、スクリプト解釈手段33からアプリケーション11aに対する起動要求が発生すると、リモート管理手段32内の要求処理手段323に、アプリケーション11aに対する起動要求が入力される(ステップS11)。

【0063】続いて、起動対象となるアプリケーション11aが存在するか否かを判定し(ステップS12)、ローカルサーバ3内にアプリケーション11aが存在しない(すなわち、NO)と判定されれば、アプリケーションダウンロード手段324は、要求処理手段323からの依頼を受け、センターサーバ情報321に基づいてセンターサーバ1にアクセスする(ステップS13)。

【0064】このように、アプリケーション情報322に基づいて、センターサーバ1から起動対象のアプリケーション11aをダウンロードする(ステップS14)。なお、アプリケーション情報322は、ローカルサーバ3内でなくセンターサーバ1内にあってもよい。

【0065】こうして、アプリケーション11aがローカルサーバ3内に存在する状態になると、要求処理手段323は、アプリケーション11aの起動をかける(ステップS15)。

【0066】なお、ステップS12において、ローカルサーバ3内に起動対象のアプリケーション11aが存在する(すなわち、YES)と判定されれば、直ちにステップS15に進む。

【0067】また、ステップS15に続いて、アプリケーション11aの処理を記述したスクリプト12aをセンターサーバ1から読み込み、このスクリプト12aをスクリプト解釈手段33に送付して、アプリケーション11aに処理を実行させる(ステップS16)。

【0068】実際のアプリケーション11aの動作は、スクリプト12aの内容にしたがって実行される。

【0069】次に、アプリケーション11aの処理が完了したか否かを判定し(ステップS17)、処理が完了していない(すなわち、NO)と判定されれば、ステップS16の処理を繰り返す。

【0070】また、ステップS17において、アプリケーション11aの処理が完了した(すなわち、YES)と判定されれば、続いて、アプリケーション11aが常駐型か否(非常駐型)かを判定する(ステップS18)。

【0071】ステップS18において、アプリケーション11aが非常駐型である(すなわち、NO)と判定さ

れれば、アプリケーション11aを削除し(ステップS19)、図4の処理ルーチンを終了する。

【0072】また、ステップS18において、アプリケーション11aが常駐型である(すなわち、YES)と判定されれば、直ちに図4の処理ルーチンを終了し、アプリケーション11aをそのままメモリ内にとどめておく。

【0073】以上のように、リモート管理手段32において、効率的なソフトウェアのダウンロードが実行可能になる。また、非常駐型のアプリケーション11を動作終了毎に削除することにより、メモリ容量の少ないローカルサーバ3においても、処理の制約を受けずにシステムの運用を実行することができる。

【0074】実施の形態3。なお、上記実施の形態1では、ローカルサーバ3内のスクリプト解釈手段33について詳述しなかったが、スクリプト解釈手段33を図5～図7のように構成してもよい。

【0075】以下、図5～図7を参照しながら、スクリプト解釈手段33を具体化したこの発明の実施の形態3について説明する。図5はローカルサーバ3内のスクリプト解釈手段33の機能を具体的に示すブロック構成図であり、スクリプト解釈手段33の周辺構成のみが示されている。

【0076】なお、図5に示されない構成は、前述(図1、図3参照)と同様である。図5において、スクリプト解釈手段33は、スクリプト定義331、イベント一覧332、解釈手段333およびイベント駆動手段334を備えている。

【0077】解釈手段333は、スクリプト定義331を含み、スクリプト定義331にしたがってスクリプト12を解釈し、スクリプト12の定義内容に相当するイベントを出力する。

【0078】イベント駆動手段334は、イベント一覧332を含み、イベントを取り込むとともに、イベント一覧332にしたがって、イベントにより駆動される処理を抽出する。

【0079】図6はこの発明の実施の形態3によるスクリプト定義331の一例を示す説明図であり、図7はこの発明の実施の形態3によるイベント一覧332の一例を示す説明図である。

【0080】図6において、スクリプト定義331は、スクリプト12a内の各定義内容に相当するイベントをマップ形式で格納しており、たとえば、定義内容<スクリプトa>に相当するイベントAが示されている。

【0081】図7において、イベント一覧332は、各イベントに相当するアプリケーションおよび処理部をマップ形式で格納しており、たとえば、イベントAに相当するアプリケーション11aおよび処理部Aが示されている。

【0082】次に、図8のフローチャートを参照しなが

ら、図5～図7に示したこの発明の実施の形態3による動作について説明する。ここでは、スクリプト解釈手段33の処理のみに注目して説明する。

【0083】まず、ローカルサーバ3は、リモート管理手段32(図1、図3参照)の介在により、センターサーバ1内のスクリプト12aをスクリプト解釈手段33内の解釈手段333に読み込む(ステップS21)。

【0084】次に、解釈手段333は、スクリプト定義331にしたがって、定義内容<スクリプトa>に相当するイベントAを抽出する(ステップS22)。

【0085】続いて、イベント駆動手段334は、イベント一覧332にしたがって、イベントAに対応する処理(アプリケーション11a、処理A)を抽出し(ステップS23)、アプリケーション11aに対して、当該処理の実行を要求する(ステップS24)。

【0086】この結果、スクリプト解釈手段33において、安全且つ確実に処理を実行するためのスクリプト12の仕様を柔軟に定義することができる。すなわち、スクリプト定義331によって、処理とは独立する形式でスクリプト12を定義することができる。

【0087】また、スクリプト12自体は、システムを構成するネットワーク、ハードウェアおよびソフトウェアなどの仕様から制約を受けることはなく、純粹に処理の意味論的な定義を行うように規定することができる。

【0088】なお、スクリプト定義331は、上記意味論的に規定されたスクリプト12と実際のシステムとの橋渡しを行うもので、スクリプト12内の記述とシステムの仕様に基づいたイベントとの対応を定義する。

【0089】一方、イベント一覧332は、システムの仕様に基づいたイベントと実際に動作するソフトウェアの実装との橋渡しを行うもので、上記構成によりソフトウェアの設計の自由度を広げることができ、或るイベントに対応したソフトウェアの構成を自由に組むことができる。

【0090】たとえば、メモリ容量の少ないローカルサーバ3においては、メモリ消費が少なくなる実装を行い、メモリ容量の制約よりも処理速度が要求されるようなローカルサーバ3においては、性能重視の実装を行うことができる。

【0091】また、上記実装の使い分けは、ローカルサーバ3内のイベント一覧332を変更することのみによって、容易に行うことができる。

【0092】実施の形態4。なお、上記実施の形態3では、スクリプト12およびスクリプト定義331の記述形式について詳述しなかったが、それぞれ、図9および図10のように記述してもよい。

【0093】以下、図9および図10を参照しながら、スクリプト12およびスクリプト定義331の記述形式を具体化したこの発明の実施の形態4について説明する。図9および図10はスクリプト12およびスクリプト定義331のサービス記述例を示す説明図である。

【0094】図9において、スクリプト解釈手段33(図5参照)内で扱われるスクリプト12は、XML(eXtensible Markup Language)により記述されている。

【0095】図10において、解釈手段333内のスクリプト定義331は、上記XML文書に対応して、DTD(Document Type Definition)により記述されている。

10 【0096】図9のように、或るアプリケーション11の動作を記述するスクリプト12がXMLで与えられた場合、スクリプト解釈手段33においては、汎用のXMLバーサが使用される。

【0097】このとき、バーサの標準仕様であるSAX(Simple API for XML)を用いれば、タグの開始終了などのイベントを抽出することができる。

20 【0098】図9および図10のように記述されたエアコン制御サービスの場合、エアコン制御開始、初期化開始、エアコン初期化開始、温度設定、風速設定、エアコン初期化終了、温度センサ初期化開始、…というイベントが順次抽出される。

【0099】また、スクリプト解釈手段33内のイベント駆動手段334は、イベント一覧332に基づいて、対応する処理を実行していく。

【0100】以上のように、発生させるイベント毎に、XMLのタグとして定義することにより、スクリプト12(XMLドキュメント)から標準のXMLバーサを用いてイベントを抽出することができる。

30 【0101】また、解釈手段333内のスクリプト定義331を、XMLに対応したDTD形式(図10参照)で記述することにより、XMLバーサの機能に基づいて、XMLドキュメント(スクリプト12)の正当性検証を実行することができる。

【0102】すなわち、DTDの検証により、スクリプト12が正しく記述されていることを保証することができる。

40 【0103】実施の形態5。なお、上記実施の形態1では、センターサーバ1内の障害対策手段13およびローカルサーバ3内の高信頼性手段34について詳述しなかったが、障害対策手段13および高信頼性手段34を図11のように構成してもよい。

【0104】以下、図11を参照しながら、障害対策手段13および高信頼性手段34を具体化したこの発明の実施の形態5について説明する。図11は障害対策手段13および高信頼性手段34の機能を具体的に示すプロック構成図であり、図示されない構成は前述(図1参照)と同様である。

50 【0105】図11において、障害対策手段13は、障害発生時の障害情報をローカルサーバ3から取得する障

害情報取得手段131と、障害発生時の障害対策を決定してローカルサーバ3に障害対策を通知する障害対策通知手段132とを備えている。

【0106】高信頼性手段34は、障害情報341、イベント情報342、障害検知手段343、障害情報収集手段344、障害通知手段345、復旧手段346およびイベント収集手段347を備えている。

【0107】障害検知手段343は、障害の発生を検知し、障害情報収集手段344は、障害発生時の障害情報341を収集し、障害通知手段345は、障害情報341をセンターサーバ1に通知する。

【0108】復旧手段346は、センターサーバ1からの障害対策に応じて障害を復旧させる。イベント収集手段347は、イベント情報342を収集記録する。

【0109】次に、図12のフローチャートを参照しながら、この発明の実施の形態5による動作について説明する。ここでは、センターサーバ1内の障害対策手段13と、ローカルサーバ3内の高信頼性手段34との連携処理に注目して説明する。

【0110】図12において、まず、アプリケーション11またはシステム（ローカルサーバ3のOSなど）からイベントが発生すると（ステップS31）、高信頼性手段34内のイベント収集手段347は、発生したイベントを収集し、このイベントが障害関連イベントか否かを判定する（ステップS32）。

【0111】ステップS32において、収集されたイベントが通常イベントであって、障害関連（異常時）のイベントではない（すなわち、NO）と判定されれば、そのイベントをイベント情報342として記録して（ステップS33）、図12の処理ルーチンを終了する。

【0112】一方、ステップS32において、収集されたイベントが障害関連（異常時）のイベントである（すなわち、YES）と判定されれば、障害が発生したことを障害検知手段343に通知する（ステップS34）。

【0113】障害検知手段343は、異常時のイベントを調べて障害の種類（障害情報）を特定するとともに、障害情報収集手段344を起動して障害情報341として記録する（ステップS35）。

【0114】こうして検知された障害情報341は、続いて、障害通知手段345を介してセンターサーバ1に送られる（ステップS36）。これに応答して、センターサーバ1内の障害対策手段13は、以下の処理を実行する。

【0115】すなわち、障害情報取得手段131は、ローカルサーバ3から到着した障害情報341を受け、障害対策通知手段132は、障害情報341に対する適切な障害対策を決定してローカルサーバ3に送信する（ステップS37）。

【0116】これに応答して、ローカルサーバ3内の高信頼性手段34は、復旧手段346による処理を実行す

る。すなわち、センターサーバ1から送られてきた障害対策に基づいて、障害からの復旧作業を実行し（ステップS38）、図12の処理ルーチンを終了する。

【0117】このように、センターサーバ1内の障害対策手段13とローカルサーバ3内の高信頼性手段34との間の連携処理により、異常が発生してもシステム全体の動作不全を未然に防止することができる。

【0118】また、ログとしてローカルサーバ3内に蓄えられた障害情報341およびイベント情報342は、後々のシステム解析などに利用することができ、万一システム全体が停止した場合でも、その解析をスムーズに行うことができる。

【0119】なお、障害情報341は、センターサーバ1に送られるので、ローカルサーバ3内でなく、センターサーバ1内で保存されてもよい。

【0120】実施の形態6。なお、上記実施の形態5では、障害対策手段13内の障害対策通知手段132について詳述しなかったが、障害対策通知手段132を図13のように構成してもよい。

【0121】以下、図13を参照しながら、障害対策手段13内の障害対策通知手段132を具体化したこの発明の実施の形態6について説明する。図13は障害対策通知手段132の機能を具体的に示すブロック構成図であり、図示されない構成は前述（図1、図11参照）と同様である。

【0122】図13において、センターサーバ1内の障害対策手段13は、前述の障害情報取得手段131および障害対策通知手段132に加えて、障害対策一覧表133を備えている。

【0123】障害対策一覧表133は、障害情報の種別と障害対策とを対応付けて、障害情報の種別毎の障害対策をマップ形式で格納している。

【0124】障害対策通知手段132は、障害対策検出手段1321と、通知手段1322とを備えている。

【0125】障害対策検出手段1321は、障害情報の種別に対応した対策方式を障害対策一覧表133に基づいて検出する。通知手段1322は、障害対策一覧表133から得られた障害対策を、ローカルサーバ3に通知する。

【0126】次に、図14のフローチャートを参照しながら、この発明の実施の形態6による動作について説明する。ここでは、センターサーバ1内の障害対策処理に注目して説明する。

【0127】図14において、ステップS36およびS38は前述（図12参照）と同様の処理であり、図14に示されない処理ステップS31～S35は、図12と同様である。

【0128】前述のように、ローカルサーバ3で障害が発生すると、障害通知手段345を介して、センターサーバ1に障害情報（種別）が送られてくる（ステップS

36)。

【0129】センターサーバ1内の障害情報取得手段131は、ローカルサーバ3からの障害情報を受けて（ステップS371）、これを障害対策通知手段132内の障害対策検出手段1321に渡す。

【0130】障害対策検出手段1321は、障害対策一覧表133に基づいて、送られてきた障害情報の種別に対応する障害対策を検出する（ステップS372）。

【0131】続いて、対策通知手段1322は、検出された障害対策をローカルサーバ3内の復旧手段346に送信する（ステップS373）。これにより、復旧手段346は、障害対策に対応した処置を実行する（ステップS38）。

【0132】このように、センターサーバ1内の障害対策手段13において、障害対策一覧表133に基づく障害対策検知手段1321を設けることにより、あらかじめ障害に対応した設計が行い易くなり、システム全体の予防保全が可能となる。

【0133】また、障害対策一覧表133は、更新可能なので、新たに障害情報の種別が増えた場合や、より効率的な障害対策が提案された場合にも、リアルタイムに対応することができ、直ちにシステムに反映させることができる。

【0134】実施の形態7. なお、上記実施の形態5、6では、センターサーバ1内の障害対策手段13に障害対策通知手段132を設け、障害情報の種別に応じた障害対策をローカルサーバ3に通知したが、ローカルサーバ3内の高信頼性手段34において自律的に障害対策を施してもよい。

【0135】以下、図15を参照しながら、高信頼性手段34において自律的に障害対策を施すようにしたこの発明の実施の形態7について説明する。図15は高信頼性手段34の機能を具体的に示すブロック構成図であり、前述（図1、図11、図13参照）と同様のものについては、同一符号を付して詳述を省略する。

【0136】図15において、センターサーバ1内の障害対策手段13は、障害情報取得手段131を備えているものの、前述（図11、図13参照）の障害対策通知手段132が削除された構成となっている。

【0137】ローカルサーバ3内の高信頼性手段34は、前述の障害情報341、イベント情報342、障害検知手段343、障害情報収集手段344およびイベント収集手段347に加えて、復旧手段348および通知手段349を備えている。

【0138】復旧手段348は、障害発生時の障害への対応を自律的に行い、障害を自動復旧させる。通知手段349は、障害情報341および自動復旧の情報をセンターサーバ1内の障害情報取得手段131に通知する。

【0139】次に、図16のフローチャートを参照しながら、この発明の実施の形態7による高信頼性手段34

の動作について説明する。図12において、ステップS31～S35は前述（図12参照）と同様の処理である。

【0140】この場合、イベント発生（ステップS31）から障害情報の収集（ステップS35）までは前述と同様である。以下、ローカルサーバ3内の高信頼性手段34は、センターサーバ1に障害情報を通知せずに、ローカルサーバ3内で障害に対する自動復旧処置（ステップS39）を完結する。

10 【0141】最後に、障害情報341および自動復旧情報の結果のみをセンターサーバ1内の障害情報取得手段131に通知し（ステップS40）、図16の処理ルーチンを終了する。

【0142】これにより、センターサーバ1内の障害対策手段13の機能構成が簡略化されるとともに、障害情報341や障害対策の通知処理（ステップS40）が、ローカルサーバ3からセンターサーバ1への1回で済み、広域ネットワーク2に流れる通信量を低減させることができる。

20 【0143】また、障害発生時において、ローカルサーバ3内で直ちに復旧が完結するので、障害への対処時間を短縮することができる。

【0144】なお、この障害対処方式は、障害の程度が非常に軽微な場合や、障害を起こしたアプリケーション11を単に再起動するのみで済む場合など、単純な処置の場合に有効である。したがって、複雑な対処が必要となる障害への対応は、前述の実施の形態5および6で示した障害対処方式の方が有効となる場合もあり得る。

30 【0145】実施の形態8. なお、上記実施の形態1～7では、センターサーバ1に接続されるローカルサーバ3の数について言及しなかったが、センターサーバ1に複数のローカルサーバ3を接続してもよい。

【0146】以下、図17を参照しながら、センターサーバ1に複数のローカルサーバ3を接続したこの発明の実施の形態8について説明する。図18はこの発明の実施の形態8を示すブロック構成図であり、前述（図1参照）と同様のものについては、同一符号を付して詳述を省略する。

40 【0147】図17において、センターサーバ1には、たとえばn個のローカルサーバ3a～3nが接続されている。同一構成のローカルサーバ3a～3nには、それぞれ、ローカルネットワーク4a～4nを介して管理制御対象機器5a～5nが接続されている。

【0148】図17のように、複数のローカルサーバ3a～3nが存在するシステムであっても、センターサーバ1は論理的に1台のみでよく、基本的な動作は前述と同様であり、前述と同等の作用効果を奏する。

50 【0149】したがって、大規模なネットワーク指向のソフトウェア流通システムにおいても、前述と同様に信頼性を向上させることができる。

【0150】なお、システムの規模によっては、システム規模の増大に対応してセンターサーバ1の台数を増やし、1つのセンターサーバ1の負荷を軽減させることもできる。

【0151】また、複数のセンターサーバ1を設けた場合、センターサーバ機能に冗長性を持たせることができるので、1つのセンターサーバがダウンしても他のセンターサーバによりバックアップすることができ、さらに信頼性を向上させることができる。

【0152】実施の形態9。なお、上記実施の形態1では、センターサーバ1内に障害対策手段13を設け、ローカルサーバ3内に高信頼性手段34を設けたが、障害対策手段13および高信頼性手段34を省略してもよい。

【0153】以下、図18を参照しながら、障害対策手段13および高信頼性手段34を省略したこの発明の実施の形態9について説明する。図18はこの発明の実施の形態9を示すブロック構成図であり、前述(図1参照)と同様のものについては、同一符号を付して詳述を省略する。

【0154】図18において、センターサーバ1およびローカルサーバ3は、前述(図1参照)の障害対策手段13および高信頼性手段34が削除された構成となっている。

【0155】したがって、図18に示したこの発明の実施の形態9による動作は、前述(図2参照)のフローチャートから、障害対策手段13および高信頼性手段34に関連するステップS5～S8を削除したものとなる。

【0156】この場合、ローカルサーバ3内で動作するソフトウェアを非常に単純化させることができる。ただし、図18の構成は、スクリプト12による動作制御のみで十分に信頼性を確保する場合に有効である。

【0157】これにより、ローカルサーバ3のソフトウェア構成が単純化され、少ないメモリ容量のローカルサーバ3を用いても、信頼性の高い処理を実現することができる。

【0158】実施の形態10。なお、上記実施の形態9では、センターサーバ1に接続されるローカルサーバ3の数について言及しなかったが、前述の実施の形態8と同様に、センターサーバ1に複数のローカルサーバ3を接続してもよい。

【0159】以下、図19を参照しながら、センターサーバ1に複数のローカルサーバ3を接続したこの発明の実施の形態10について説明する。図19はこの発明の実施の形態10を示すブロック構成図であり、前述(図18参照)と同様のものについては、同一符号を付して詳述を省略する。

【0160】図19において、センターサーバ1には、複数のローカルサーバ3a～3nが接続されている。図

19のように、複数のローカルサーバ3a～3nを有するシステムにおいても、センターサーバ1は論理的に1台でよく、基本的な動作は前述と同様である。

【0161】これにより、単純な処理を行うローカルサーバ3が複数存在する大規模なネットワーク指向のソフトウェア流通システムにおいても、信頼性を向上させることができる。

【0162】実施の形態11。なお、上記実施の形態8(図17参照)では、センターサーバ1に接続される全てのローカルサーバ3a～3nに高信頼性手段34a～34nを設けたが、一部のローカルサーバ3のみに高信頼性手段34を設け、他のローカルサーバ3から高信頼性手段34を削除してもよい。

【0163】以下、図20を参照しながら、複数のうちの一部のローカルサーバ3のみに高信頼性手段34を設けたこの発明の実施の形態11について説明する。図20はこの発明の実施の形態11を示すブロック構成図であり、前述(図17参照)と同様のものについては、同一符号を付して詳述を省略する。

【0164】図20において、複数のローカルサーバ3a～3nのうち、一部のローカルサーバ3aは、高信頼性手段34が削除された構成となっており、他のローカルサーバ3nは、高信頼性手段34nを備えている。

【0165】図20に示したこの発明の実施の形態11による基本的動作は、各ローカルサーバ3a～3nに関して前述と同様である。

【0166】図20の構成は、高信頼性手段34nが必要なローカルサーバ3nと、高信頼性手段34が不要なローカルサーバ3aとが混在するシステムにおいて、有効である。

【0167】この場合も、障害対策手段13を備えたセンターサーバ1が論理的に1台あればよく、基本的な動作は前述と同様である。また、システム規模の増大に応じてセンターサーバ1の台数を増やしてもよい。

【0168】これにより、単純な処理を行うローカルサーバ3aと、通常の高信頼性処理を行うローカルサーバ3nとが複数混在する大規模なネットワーク指向のソフトウェア流通システムにおいても、信頼性を向上させることができる。

【0169】実施の形態12。なお、上記実施の形態1では、センターサーバ1にスクリプト12を設け、ローカルサーバ3にスクリプト解釈手段33を設けたが、スクリプト12およびスクリプト解釈手段33を削除してもよい。

【0170】この場合、アプリケーション11のみで動作可能なシンプルなシステムにおいて有効であり、センターサーバ1およびローカルサーバ3の構成を簡略化することができる。

【0171】また、障害発生時の障害対策については、前述と同様の作用効果を奏する。また、センターサーバ

1に複数のローカルサーバ3が接続されたシステムにも適用可能である。

【0172】

【発明の効果】以上のように、この発明の請求項1によれば、センターサーバと、広域ネットワークを介してセンターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、センターサーバは、ローカルサーバにダウンロードさせて動作するアプリケーションと、アプリケーションの動作を記述したスクリプトと、障害発生時の対応を行う障害対策手段とを備え、ローカルサーバは、ネットワーク指向言語実行環境と、アプリケーションのセンターサーバからのダウンロードや処理終了後の削除などを行うリモート管理手段と、スクリプトを解釈してアプリケーションに処理を要求するスクリプト解釈手段と、アプリケーションの実行中に発生するイベント情報の記録や、障害発生時の情報管理および復旧処理を行う高信頼性手段とを備え、広域ネットワークを介してダウンロードされたソフトウェアの安全確実な動作を保証するとともに、万一異常が発生しても障害情報収集や復旧を支援する機能を提供したので、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【0173】また、この発明の請求項2によれば、請求項1において、リモート管理手段は、センターサーバ情報およびアプリケーション情報と、アプリケーションの実行要求に対応する要求処理手段と、センターサーバ情報およびアプリケーション情報に基づいてアプリケーションをセンターサーバからダウンロードするアプリケーションダウンロード手段と、アプリケーション情報に基づいてアプリケーションの起動や削除などの処理を行うアプリケーション管理手段とを含むので、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【0174】また、この発明の請求項3によれば、請求項1において、スクリプト解釈手段は、スクリプト定義およびイベント一覧と、スクリプト定義にしたがってスクリプトを解釈してスクリプトの定義内容に相当するイベントを出力する解釈手段と、イベントを取り込むとともにイベント一覧にしたがってイベントにより駆動される処理を抽出するイベント駆動手段とを含むので、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【0175】また、この発明の請求項4によれば、請求項1において、スクリプトは、XML (eXtensible Markup Language) により記述され、スクリプト定義は、DTD (Document Type Definition) により記述されたので、広域ネットワーク上を流通するソフトウェアの処理

の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【0176】また、この発明の請求項5によれば、請求項1において、障害対策手段は、障害発生時の障害情報をローカルサーバから取得する障害情報取得手段と、障害発生時の障害対策を決定してローカルサーバに通知する障害対策通知手段とを含み、高信頼性手段は、障害発生時に障害の発生を検知する障害検知手段と、障害発生時の障害情報を収集する障害情報収集手段と、障害情報をセンターサーバに通知する障害通知手段と、センターサーバからの障害対策に応じて障害を復旧させる復旧手段と、イベント情報を収集記録するイベント収集手段とを含むので、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【0177】また、この発明の請求項6によれば、請求項5において、障害対策手段は、障害情報の種別毎に障害対策を格納する障害対策一覧表を含み、障害対策通知手段は、障害情報の種別に対応した障害対策を障害対策一覧表に基づいて検出する障害対策検出手段と、障害対策をローカルサーバに通知する通知手段とを含むので、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【0178】また、この発明の請求項7によれば、請求項1において、障害対策手段は、障害発生時の障害情報をローカルサーバから取得する障害情報取得手段を含み、高信頼性手段は、障害発生時に障害の発生を検知する障害検知手段と、障害発生時の障害情報を収集する障害情報収集手段と、障害発生時の障害への対応を自律的に行い障害を自動復旧させる復旧手段と、障害情報を含む自動復旧の情報をセンターサーバに通知する通知手段と、イベント情報を収集記録するイベント収集手段とを含むので、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【0179】また、この発明の請求項8によれば、請求項1において、ローカルサーバは複数存在し、それぞれ、ネットワーク指向言語実行環境、リモート管理手段、スクリプト解釈手段および高信頼性手段を備えたので、大規模なシステムに対しても、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【0180】また、この発明の請求項9によれば、請求項8において、センターサーバは複数存在し、それぞれ、アプリケーション、スクリプトおよび障害対策手段を備えたので、大規模なシステムに対しても、センターサーバの負荷を軽減するとともに冗長性をもたせ、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効

果がある。

【0181】また、この発明の請求項10によれば、請求項1において、ローカルサーバは複数存在し、少なくとも1つのローカルサーバは、ネットワーク指向言語実行環境、リモート管理手段、スクリプト解釈手段および高信頼性手段を備え、他のローカルサーバは、ネットワーク指向言語実行環境、リモート管理手段およびスクリプト解釈手段を備えたので、大規模なシステムに対しても、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【0182】また、この発明の請求項11によれば、センターサーバと、広域ネットワークを介してセンターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、センターサーバは、ローカルサーバにダウンロードさせて動作するアプリケーションと、アプリケーションの動作を記述したスクリプトとを備え、ローカルサーバは、ネットワーク指向言語実行環境と、アプリケーションのセンターサーバからのダウンロードや処理終了後の削除などを行うリモート管理手段と、スクリプトを解釈してアプリケーションに処理を要求するスクリプト解釈手段とを備えたので、ローカルサーバの機能構成を簡略化するとともに、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【0183】また、この発明の請求項12によれば、請求項11において、ローカルサーバは複数存在し、それ、ネットワーク指向言語実行環境、リモート管理手段およびスクリプト解釈手段を備えたので、大規模なシステムに対しても、ローカルサーバの機能構成を簡略化するとともに、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【0184】また、この発明の請求項13によれば、センターサーバと、広域ネットワークを介してセンターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、センターサーバは、ローカルサーバにダウンロードさせて動作するアプリケーションと、障害発生時の対応を行う障害対策手段とを備え、ローカルサーバは、ネットワーク指向言語実行環境と、アプリケーションのセンターサーバからのダウンロードや処理終了後の削除などを行うリモート管理手段と、アプリケーションの実行中に発生するイベント情報の記録や、障害発生時の情報管理および復旧処理を行う高信頼性手段とを備えたので、センターサーバの機能構成を簡略化するとともに、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【0185】また、この発明の請求項14によれば、請求項13において、ローカルサーバは複数存在し、それ、ネットワーク指向言語実行環境、リモート管理手段および高信頼性手段を備えたので、大規模なシステムに対しても、センターサーバの機能構成を簡略化するとともに、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【図面の簡単な説明】

10 【図1】この発明の実施の形態1を示すブロック構成図である。

【図2】この発明の実施の形態1による動作を示すフローチャートである。

【図3】この発明の実施の形態2を示すブロック構成図である。

【図4】この発明の実施の形態2による動作を示すフローチャートである。

【図5】この発明の実施の形態3の要部を示すブロック構成図である。

20 【図6】この発明の実施の形態3によるスクリプト定義の一例を示す説明図である。

【図7】この発明の実施の形態3によるイベント一覧の一例を示す説明図である。

【図8】この発明の実施の形態3による動作を示すフローチャートである。

【図9】この発明の実施の形態4によるサービス記述例を示す説明図である。

【図10】この発明の実施の形態4によるサービス記述例を示す説明図である。

30 【図11】この発明の実施の形態5を示すブロック構成図である。

【図12】この発明の実施の形態5による動作を示すフローチャートである。

【図13】この発明の実施の形態6を示すブロック構成図である。

【図14】この発明の実施の形態6による動作を示すフローチャートである。

【図15】この発明の実施の形態7を示すブロック構成図である。

40 【図16】この発明の実施の形態7による動作を示すフローチャートである。

【図17】この発明の実施の形態8を示すブロック構成図である。

【図18】この発明の実施の形態9を示すブロック構成図である。

【図19】この発明の実施の形態10を示すブロック構成図である。

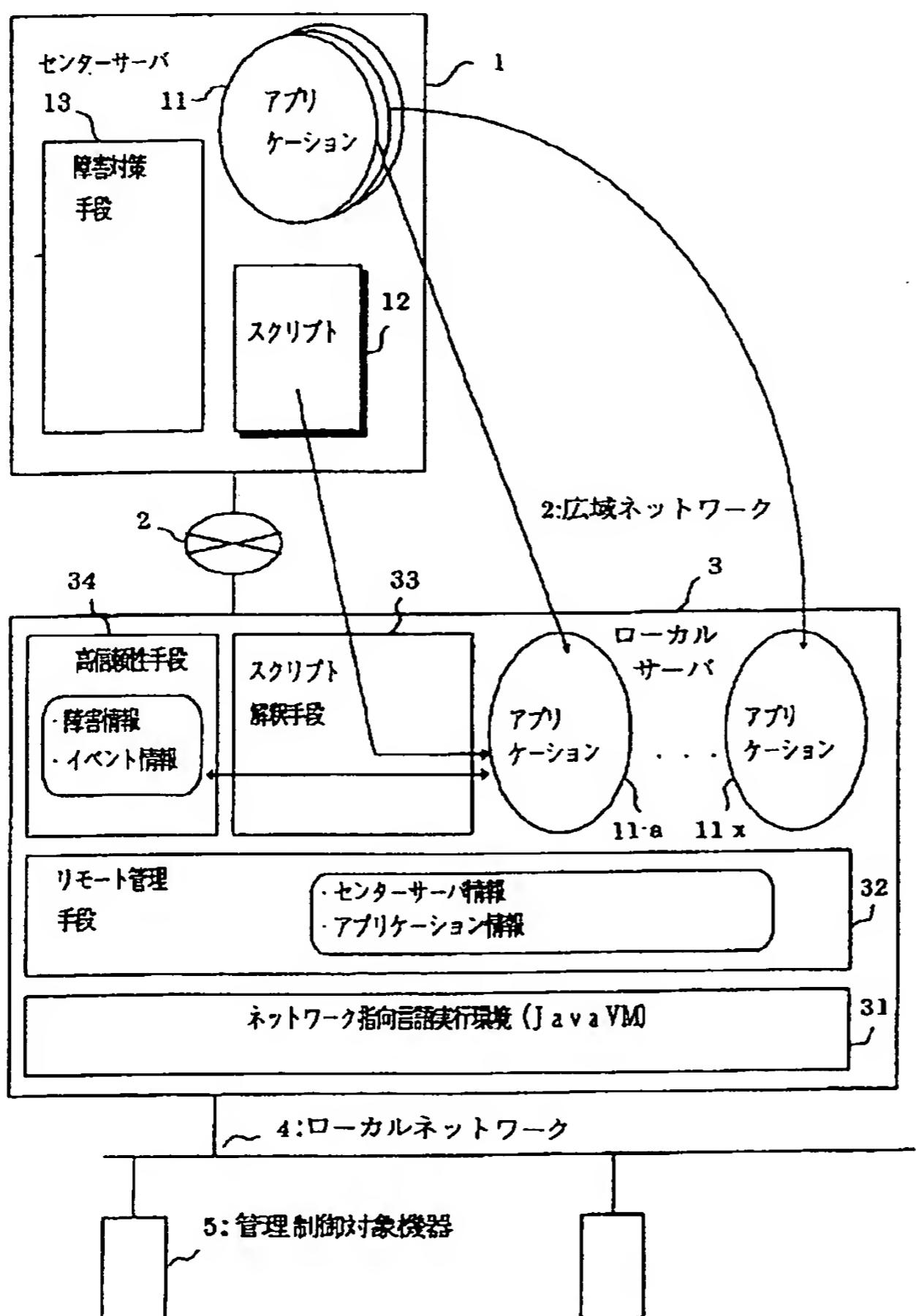
【図20】この発明の実施の形態11を示すブロック構成図である。

50 【符号の説明】

1 センターサーバ、2 広域ネットワーク、3、3a～3n ローカルサーバ、4、4a～4n ローカルネットワーク、5、5a～5n 管理制御対象機器、11、11a～11x アプリケーション、12、12a スクリプト、13 障害対策手段、31、31a～31n ネットワーク指向言語実行環境 (JavaVM)、32、32a～32n リモート管理手段、33、33a～33n スクリプト解釈手段、34、34a～34n 高信頼性手段、131 障害情報取得手段、132 障害対策通知手段、133 障害対策一覧\*10

\*表、321 センターサーバ情報、322 アプリケーション情報、323 要求処理手段、324 アプリケーションダウンロード手段、325 アプリケーション管理手段、331 スクリプト定義、332 イベント一覧、333 解釈手段、334 イベント駆動手段、341 障害情報、342 イベント情報、343 障害検知手段、344 障害情報収集手段、345 障害通知手段、346、348 復旧手段、347 イベント収集手段、349 通知手段、1321 障害対策検出手段、1322 通知手段。

【図1】



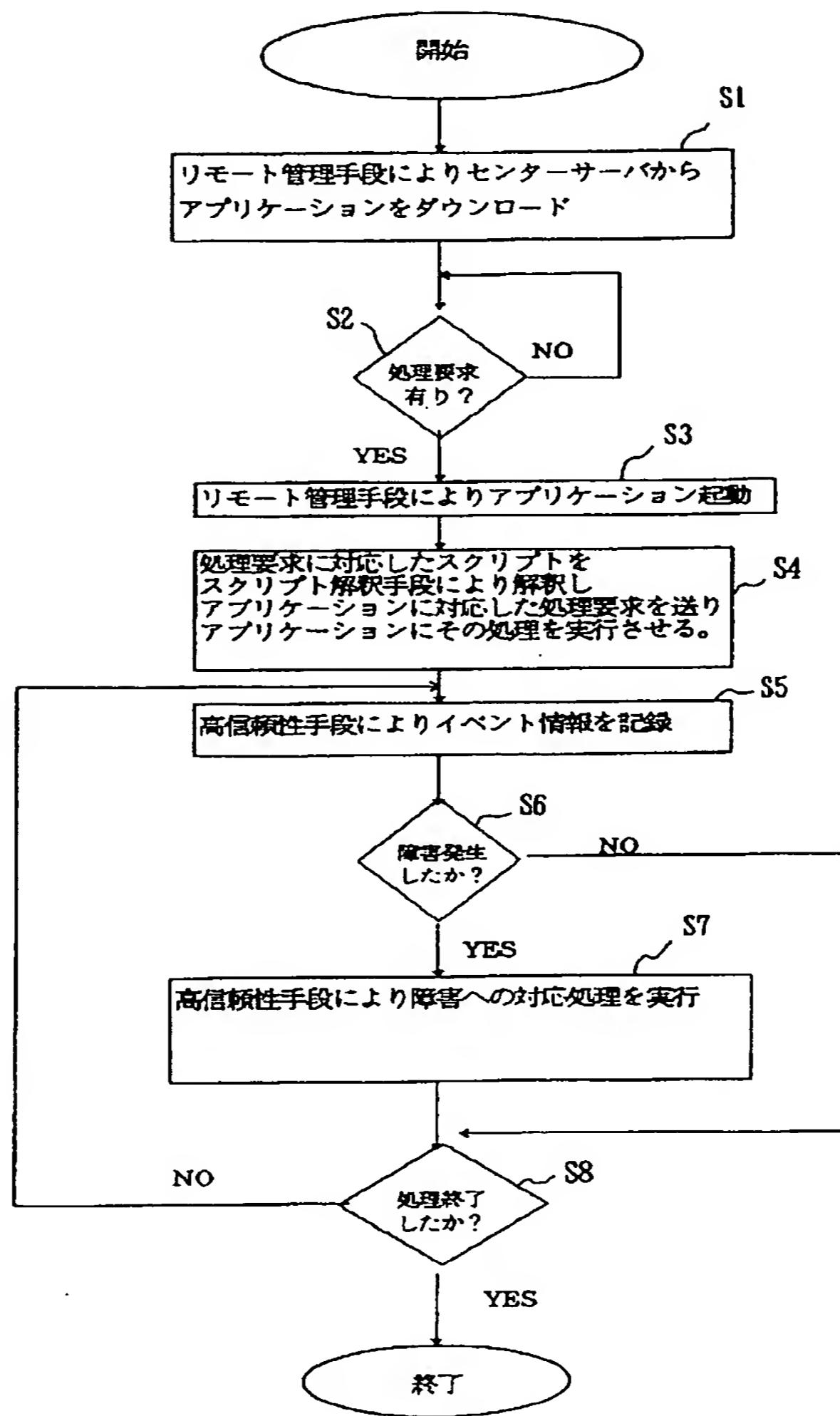
【図6】

スクリプト定義	イベント
⋮	⋮
<スクリプトa>	イベントA
⋮	⋮

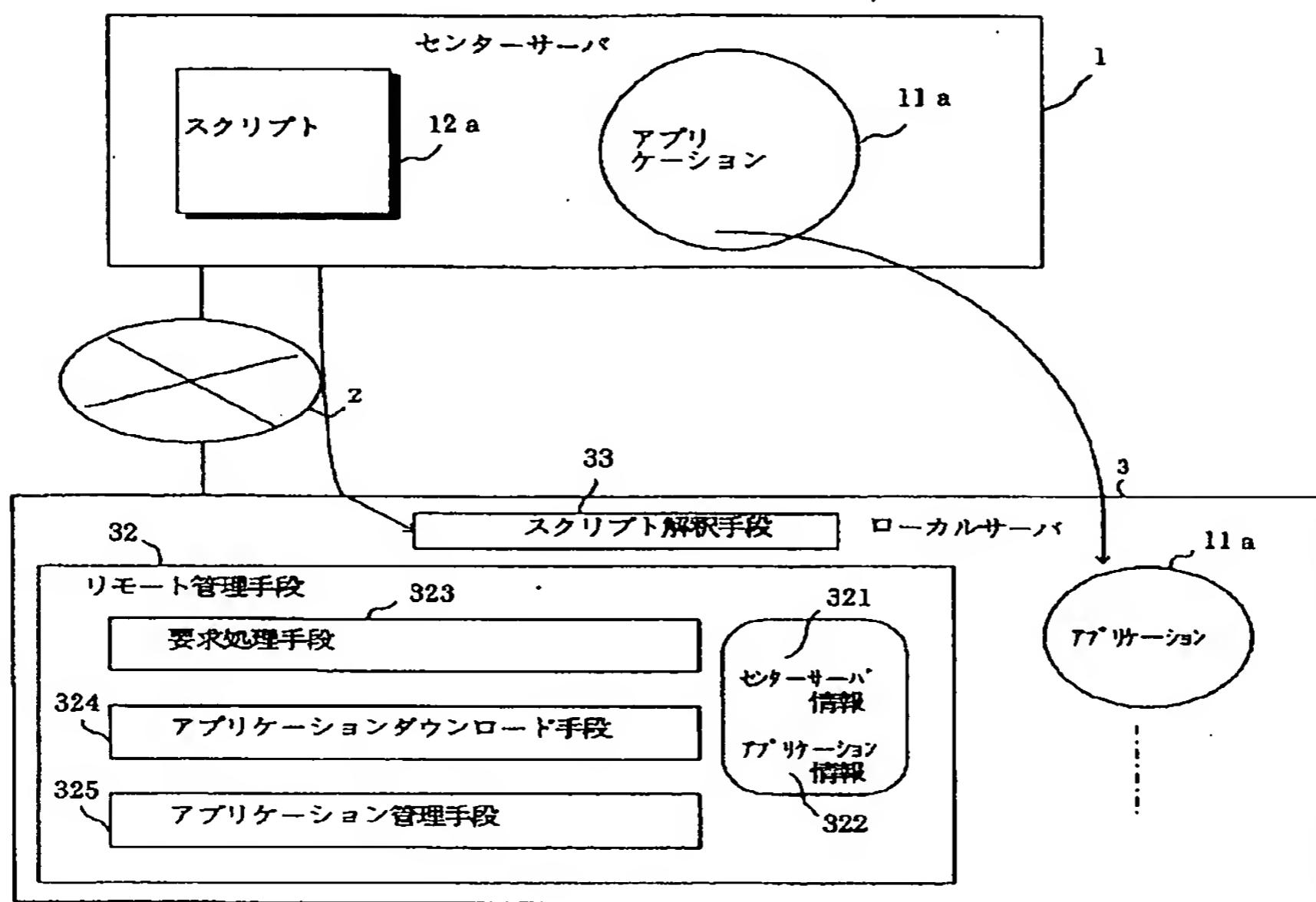
【図7】

イベント	処理
⋮	⋮
イベントA	アプリケーション11-a 処理部A
⋮	⋮

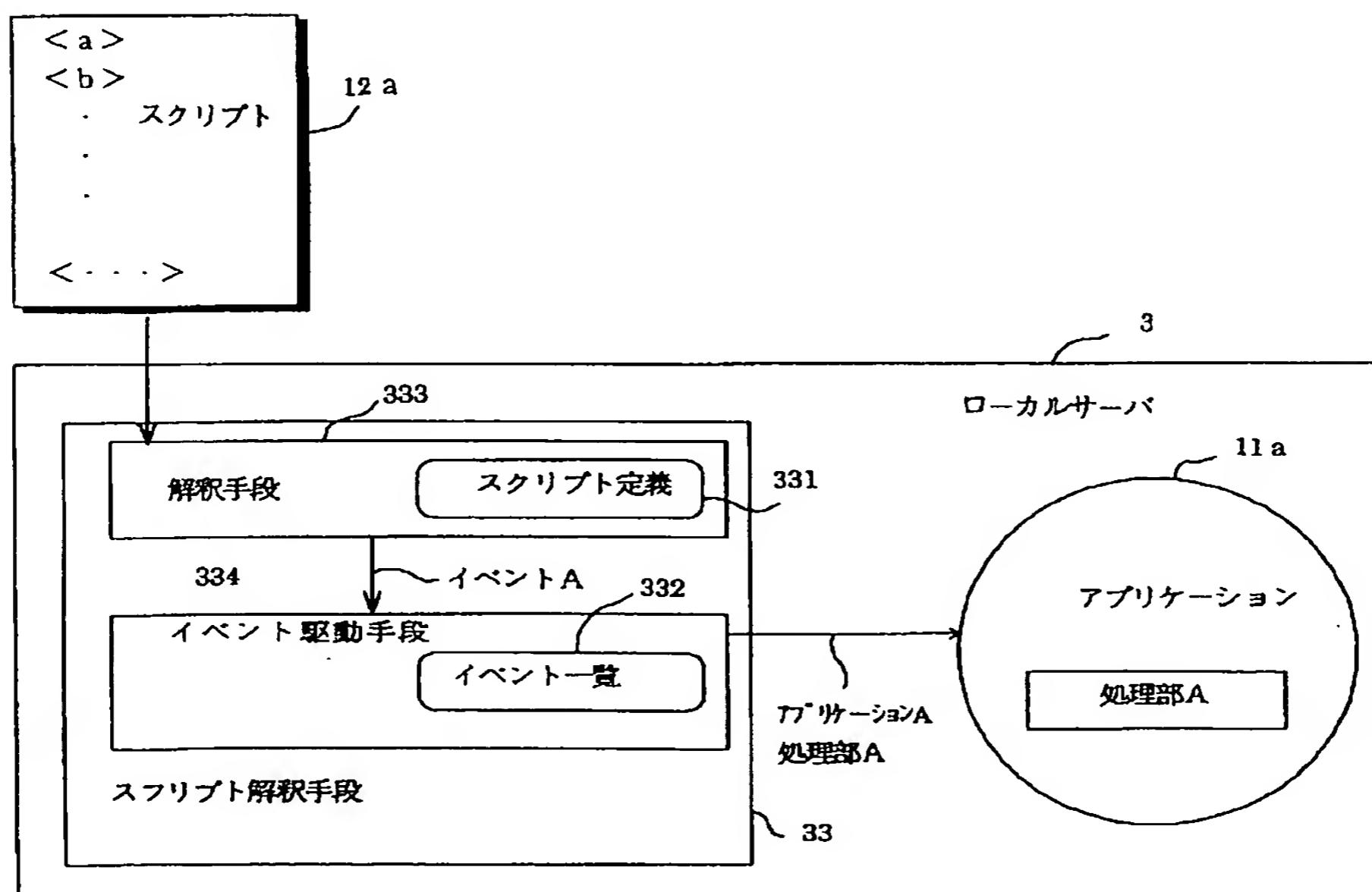
【図2】



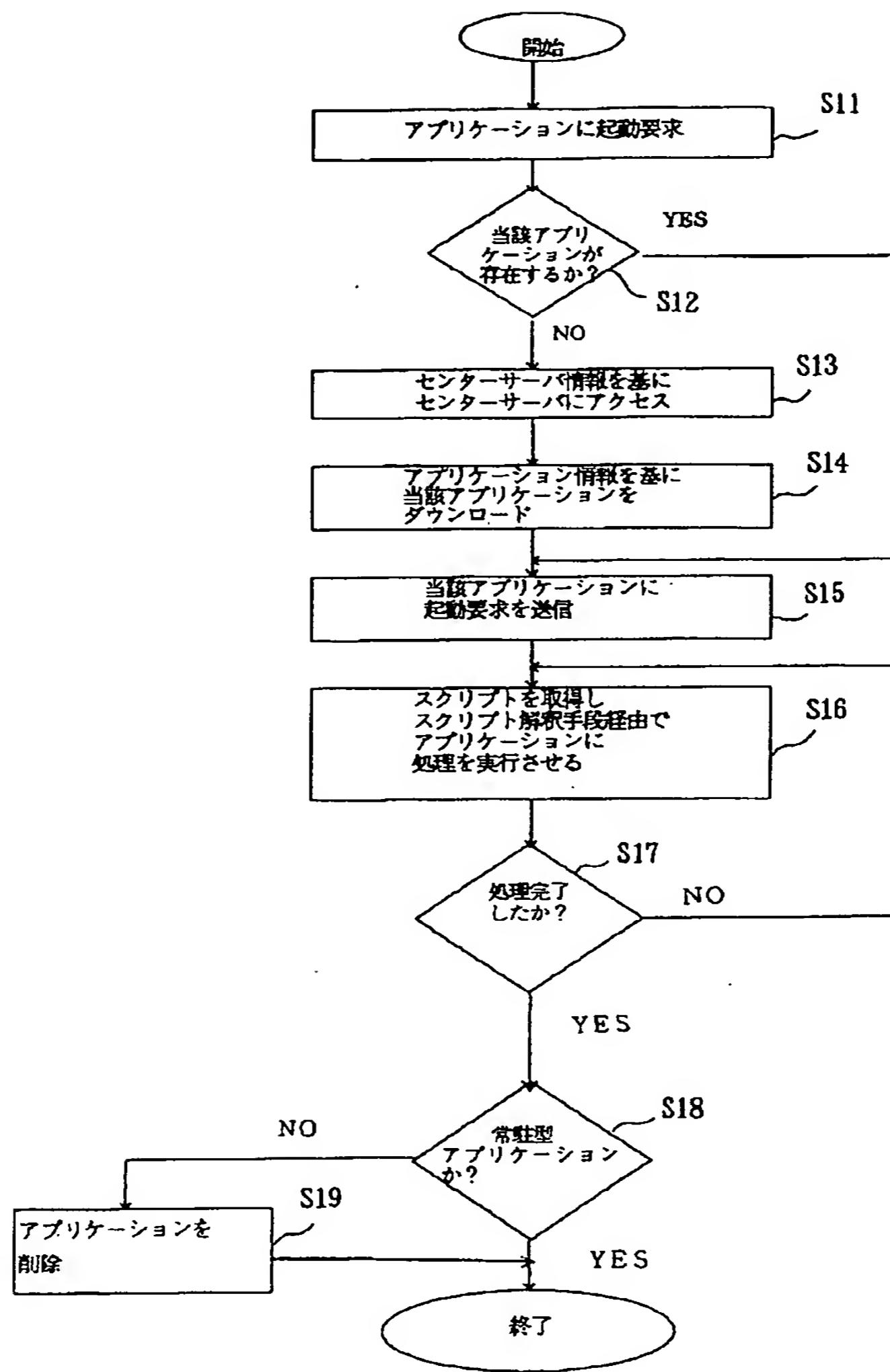
【図3】



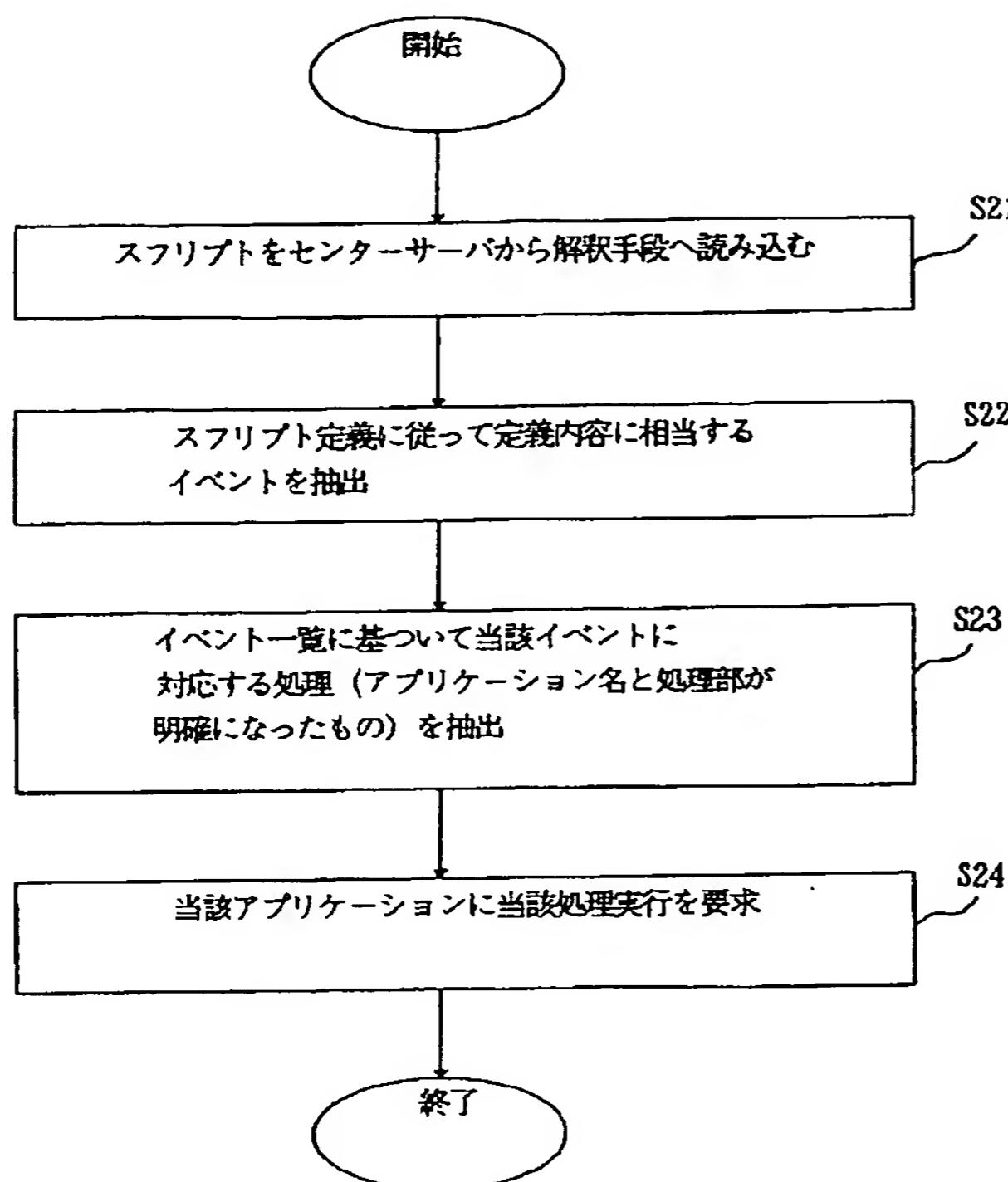
【図5】



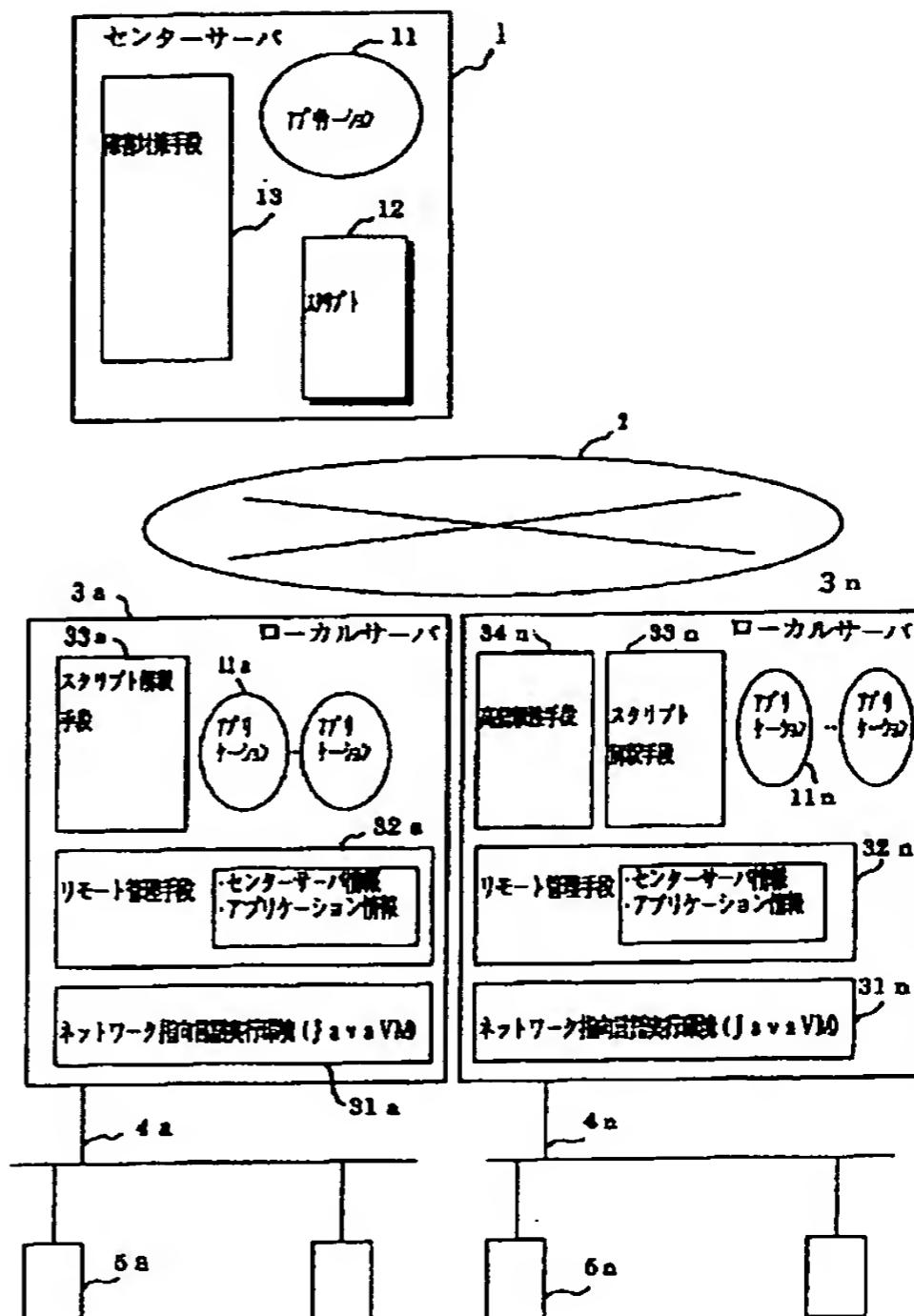
【図4】



【図8】



【図20】



【図10】

XML文書に対するDTD

```

<!DOCTYPE エアコン制御 [
<!ELEMENT エアコン制御 (初期化、自動制御) >
<!ELEMENT 初期化 (エアコン, 温度センサ) >
<!ELEMENT 自動制御 (...) >
<!ELEMENT エアコン (温度, 風速) >
<!ELEMENT センサ (...) >
<!ELEMENT 温度 (#PCDATA) >
<!ELEMENT 風速 (#PCDATA) >
<!ELEMENT 温度 単位 (摺氏 | 華氏) “摺氏” >
<!ATTRLIST 温度 単位 (摺氏 | 華氏) “摺氏” >
]
  
```

【図9】

```

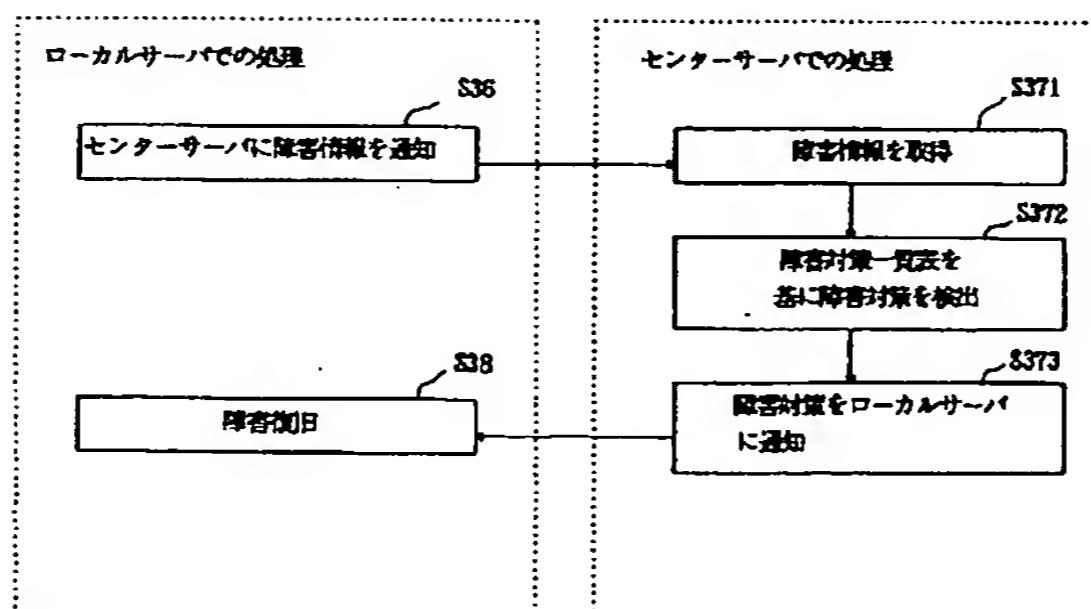
<? XML version="1.0" ?>
<エアコン制御>
<初期化>
  <エアコン>
    <温度 単位="摂氏" >28<? 温度>
    <風速>静</風速>
  </エアコン>
  <温度センサ>

  </温度センサ>
</初期化>
<自動制御>

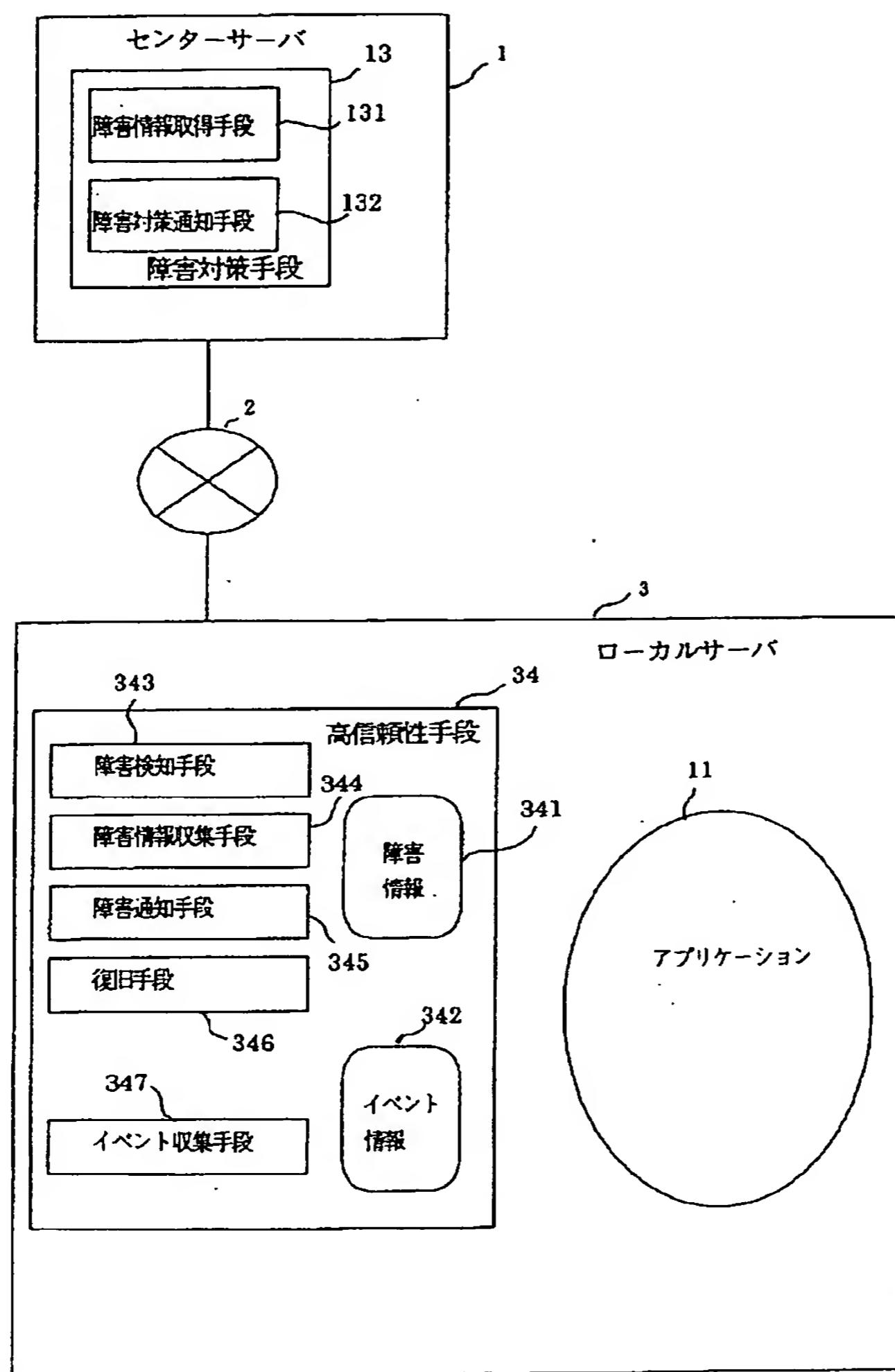
</自動制御>
</エアコン制御>

```

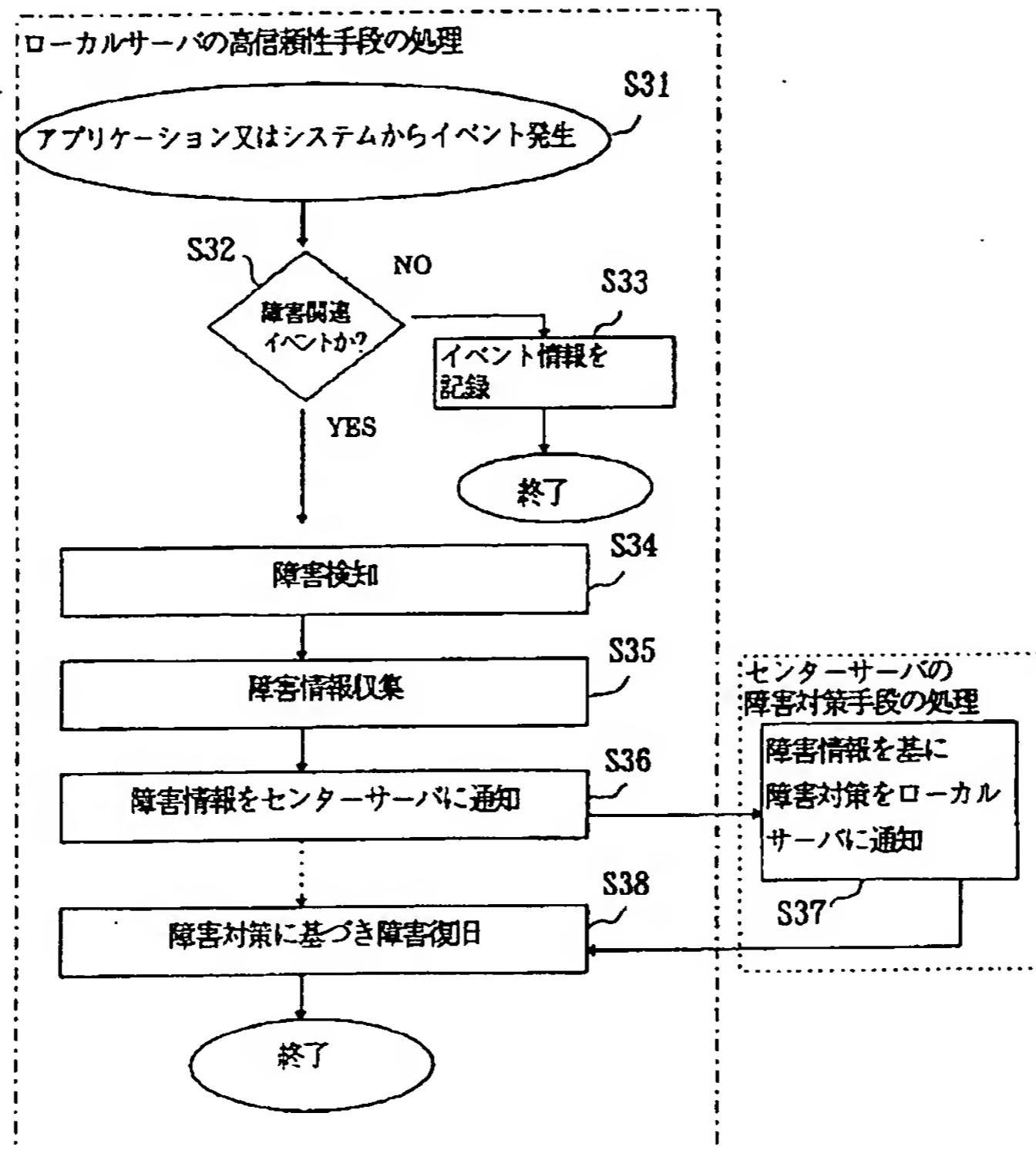
【図14】



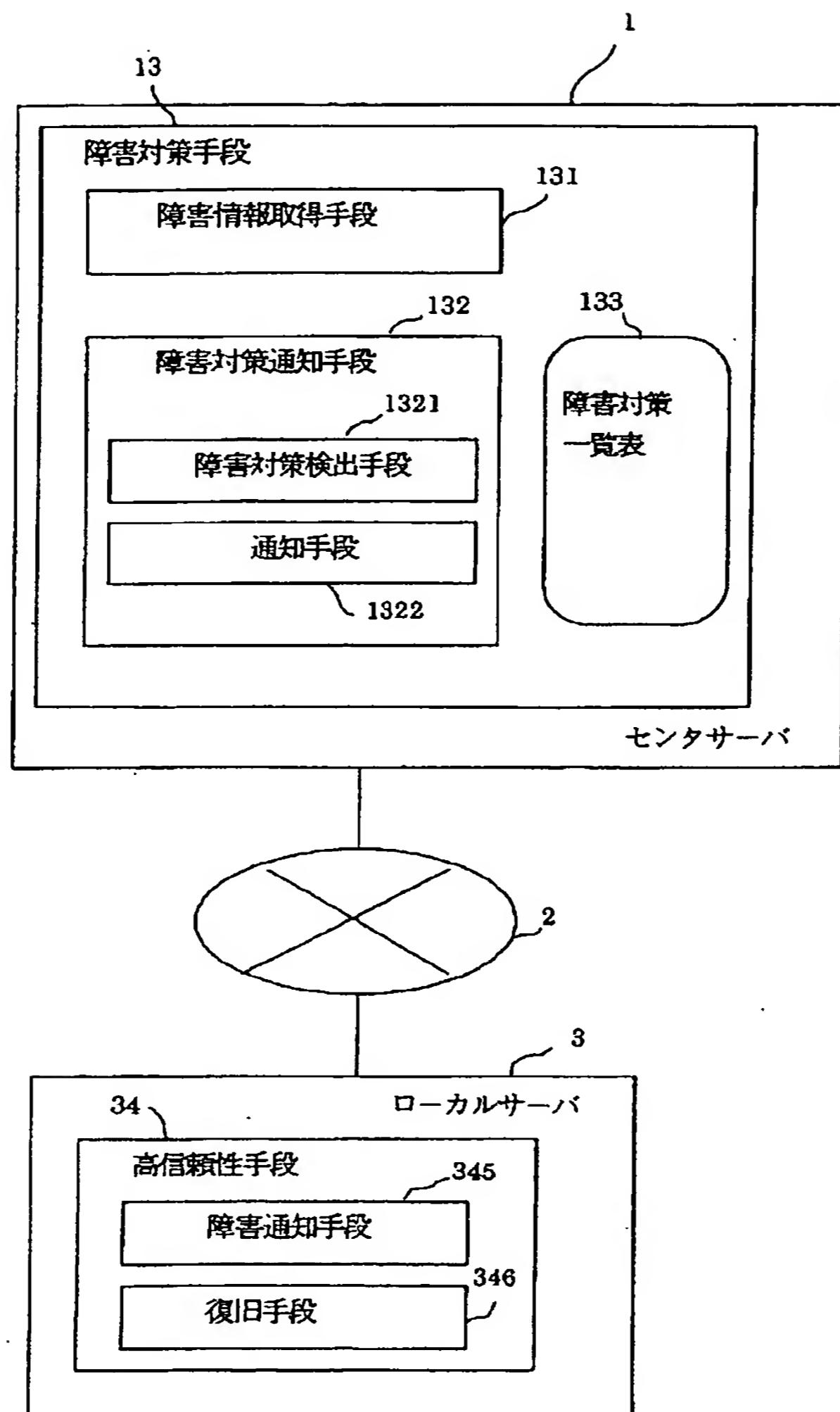
【図11】



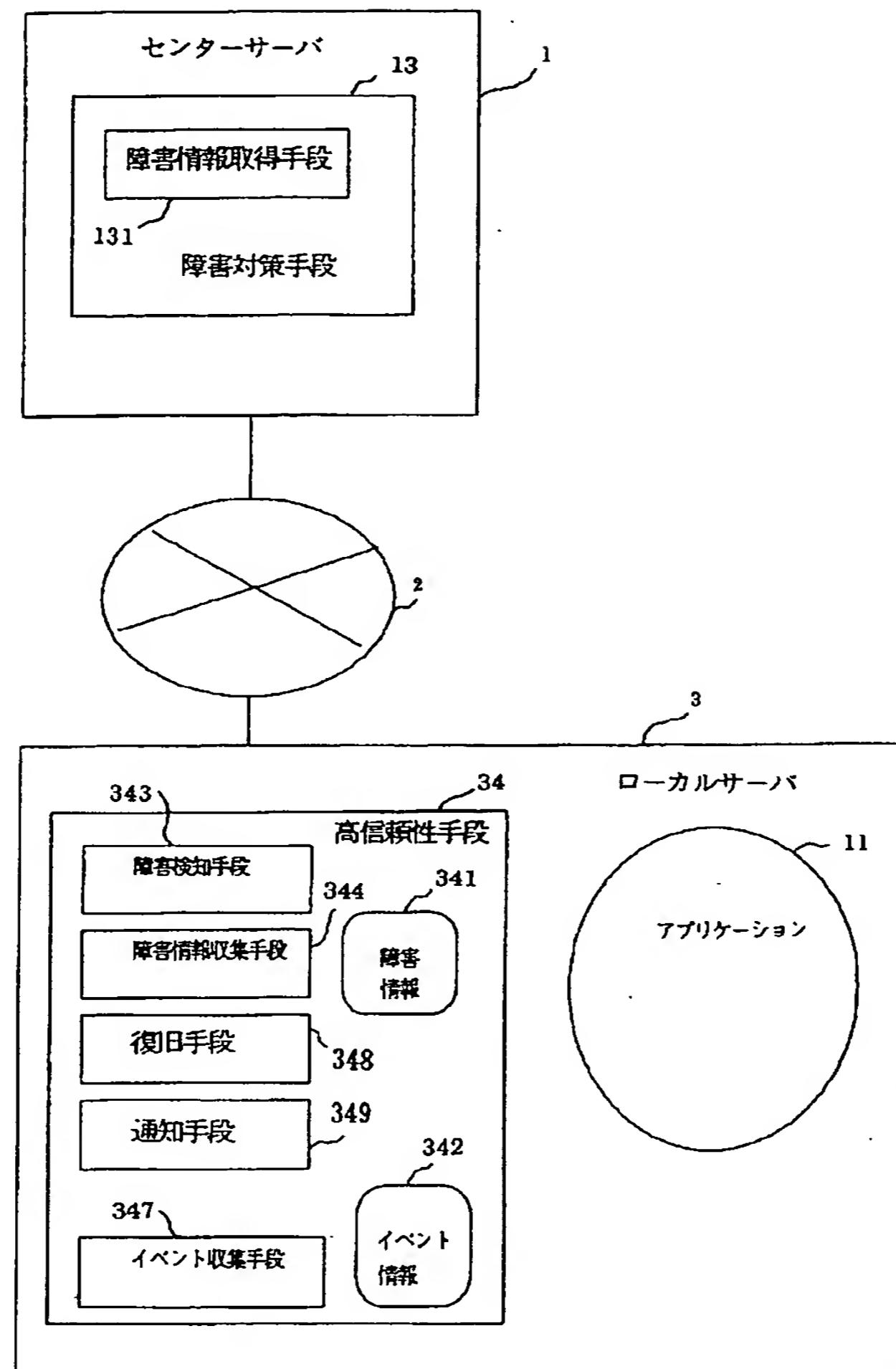
【図12】



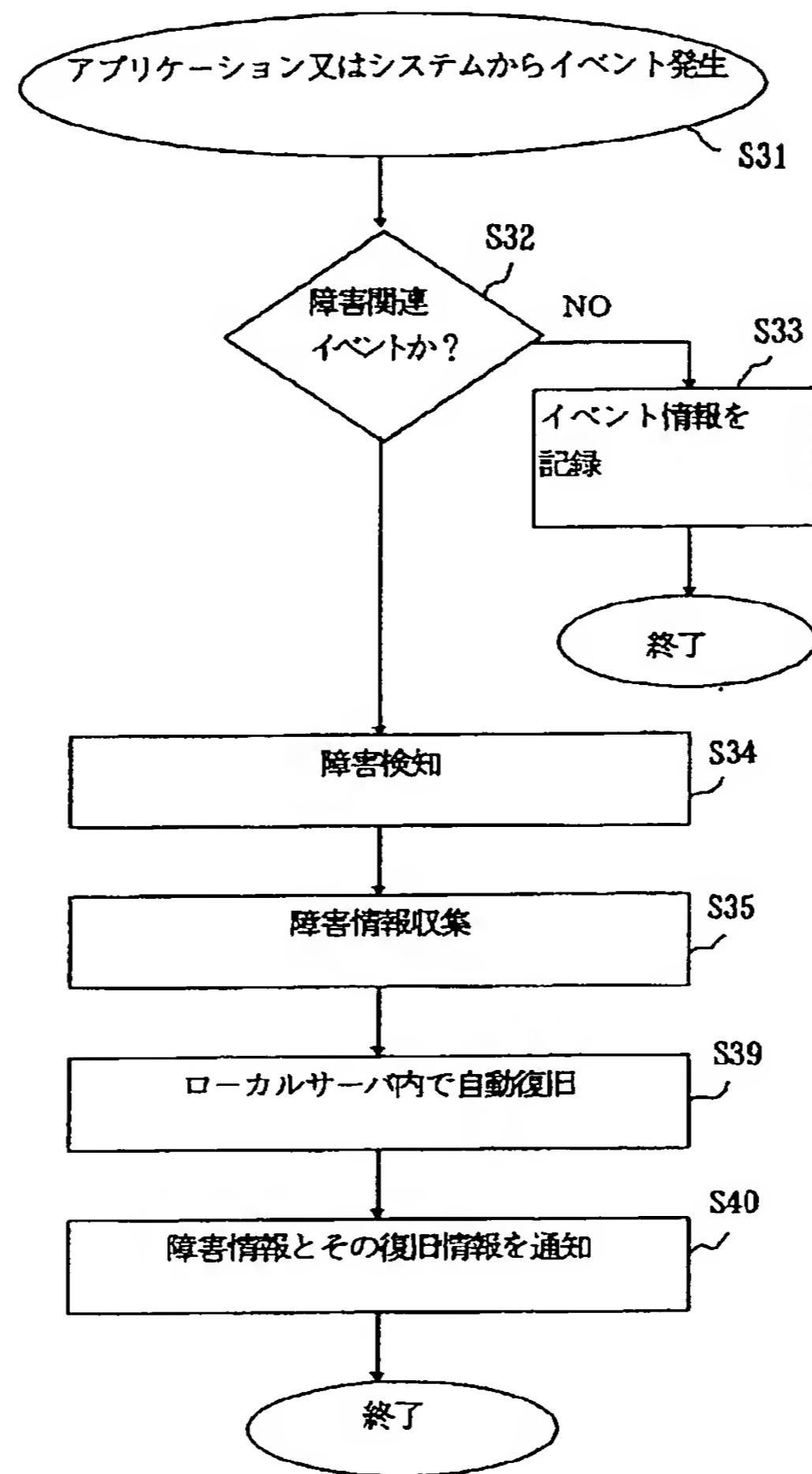
【図13】



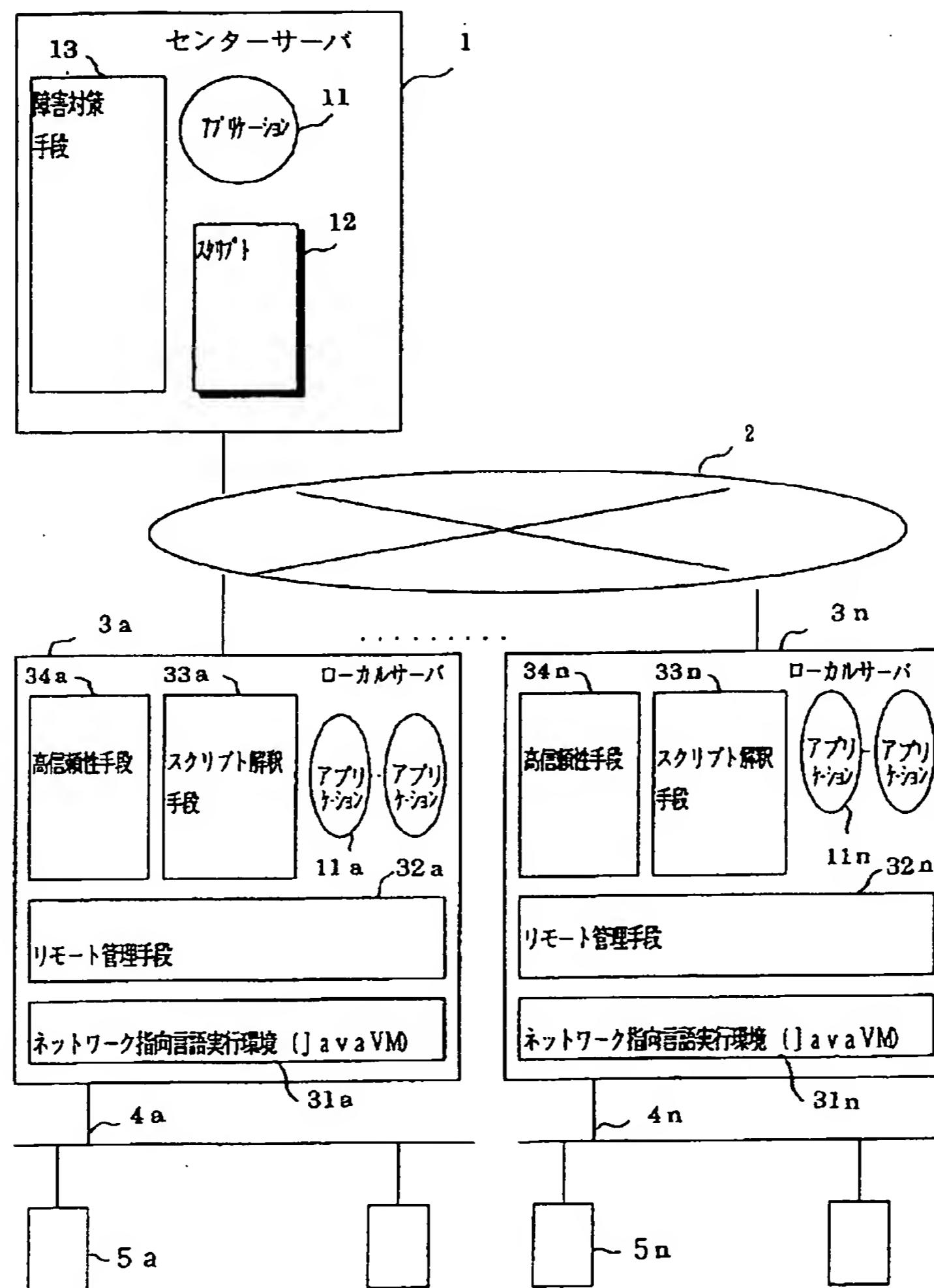
【図15】



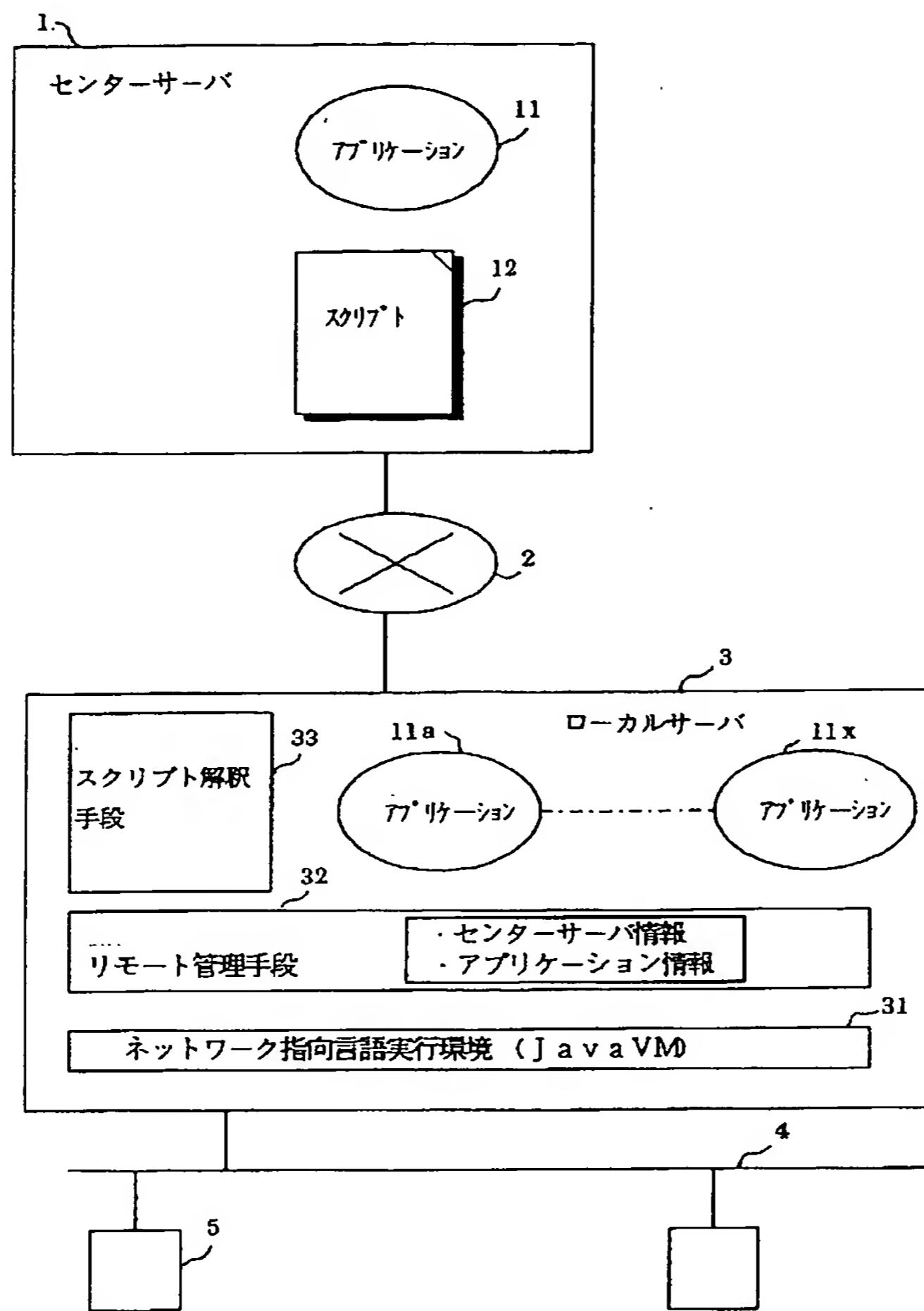
【図16】



【図17】



【図18】



【図19】

